marconi instruments

MP MASTER
MA25529

Operating Manual

H 6500

Vol. 1A

# AUTOMATIC AMPLITUDE ANALYSER
# 6500
# GPIB OPERATION

IEEE 488-1978 SH1,AH1,T5,L4,SR1,RL1,DC1,E1

© Marconi Instruments Limited 1984
Printed in the UK

Part No. 46881-558S
Print Code : A-5/84, MI 2.5c

# CONTENTS

## PRELIMINARIES

Title page
Contents

## CHAPTERS

1 General information
2 Installation
3 Operation
4 Applications

## APPENDICES

A 6500 firmware Issue 5 and pre-Issue 5 comparison
B GPIB error codes
C 6500 character set and text control codes
D Keyboard codes returned by RK command
E 6500 CRT layout chart
F Standard alphanumeric layouts for single and dual channel displays
G Digital plotter output information
H Sample digital plot

Digitized by the Internet Archive
in 2024

https://archive.org/details/manualsplus_00819

Chapter 1

# GENERAL INFORMATION

## CONTENTS

Para.
1 Introduction
1 General purpose interface bus (IEEE-488/IEC 625)
2 Interface function capabilities

## INTRODUCTION

General purpose interface bus (IEEE-488/IEC 625)

1. The 6500 GPIB interface allows the instrument to be connected to a GPIB controller and other GPIB compatible devices. This manual describes the GPIB commands implemented on 6500 and, where appropriate, gives examples of their use. Programming examples are given for use on Hewlett Packard HP85 and Series 200 controllers. Further information on the general provisions of the IEEE-488 standard and its implementation may be found in the Marconi Instruments GPIB Manual (Part No. H 54811-010P). Familiarity with 6500 LOCAL operation is assumed. See operating manual H 6500, Vol. 1 for details.

Note...

The information contained in this operating manual pertains to 6500 firmware issue 5. See Appendix A for details of compatibility with pre-issue 5 versions.

## INTERFACE FUNCTION CAPABILITIES

2. SH1 : Source Handshake - complete capability. The source handshake sequences the transmission of each data byte from the instrument over the bus data lines. The sequence is initiated when the function becomes active, and the purpose of the function is to synchronize the rate at which bytes become available to the rate at which accepting devices on the bus can receive the data.

3. AH1 : Acceptor Handshake - complete capability. The acceptor handshake sequences the reading of data bytes from the bus lines.

4. T5 : Talker Function - complete capability. The talker function provides the 6500 with the ability to send device dependent messages over the bus to other devices. Unless in TALK ONLY mode the ability to talk exists only when the instrument has been addressed as a talker. The TALK ONLY mode is provided on the 6500 to allow direct interfacing to an intelligent digital plotter.

5. L4 : Listener Function - no LISTEN ONLY function. The listener function provides a device with the ability to receive device dependent messages over the bus. The capability exists only when the device is addressed to listen by the controller.

6. SR1 : Service Request Function - complete capability. The service request function gives the 6500 the ability to interrupt the controller and request attention. The 6500 can be enabled to issue the interrupt on the occurrence of a number of a number of different events and returns a STATUS BYTE indicating the source of the interrupt when interrogated by the SERIAL POLL technique. PARALLEL POLLING is not supported.

7. RL1 : Remote/Local Function - complete capability. The remote/local function allows the 6500 to be controlled either locally by the front panel controls, or remotely by device dependent messages over the bus.

8. DC1 : Device Clear Function - complete capability. The device clear function is a general reset and may be given to all devices in the system (DCL - DEVICE CLEAR) or only to addressed devices (SDC - SELECTED DEVICE CLEAR). On receipt of DCL or SDC the 6500 performs a software reset, returning the instrument to its power-on state. The software reset includes the following actions :

SELF TEST
CLEAR ALL MEMORIES
SET DETECTOR ZERO AND NULL VOLTAGES TO NOMINAL VALUES
INITIALIZE ALL INSTRUMENT SETTINGS
READ GPIB ADDRESS SWITCH
RETURN 6500 TO LOCAL CONTROL

Note...

A self test failure may cause a GPIB interface error condition or suspend GPIB activity. The 6500 must be switched off to clear this condition.

9. E1 : Bus Driver Type. The GPIB driver devices fitted in the 6500 interface have open-collector, rather than tri-state outputs.

Chapter 2

# INSTALLATION

## CONTENTS

Para.
1 Installation procedure
2 GPIB Address/TALK ONLY Selection

Fig. Page
1 Interface installation ... ... ... ... ... ... 2
2 Label positioning ... ... ... ... ... ... ... 2
3 Address switch configuration ... ... ... ... ... 3

## INSTALLATION PROCEDURE

1. The procedure described assumes that a GPIB interface is being installed in a 6500 which previously had none fitted. The interface may be installed as follows :

(1) Switch the instrument off and disconnect from power supply.

(2) Remove and discard the rectangular cover plate from the left-hand side of the rear panel. Also remove the instrument's bottom cover. Insert the cable and p.c.b. through the rear panel such that the GPIB connector is on the left-hand side. Using the two retaining screws from the cover plate, secure the GPIB assembly to the rear panel.

(3) Feed the interface cable through the cut-out in the bulkhead metalwork as shown in Fig. 1 and connect the interface plug to SK14 on the 6500 mother board, observing correct polarity.

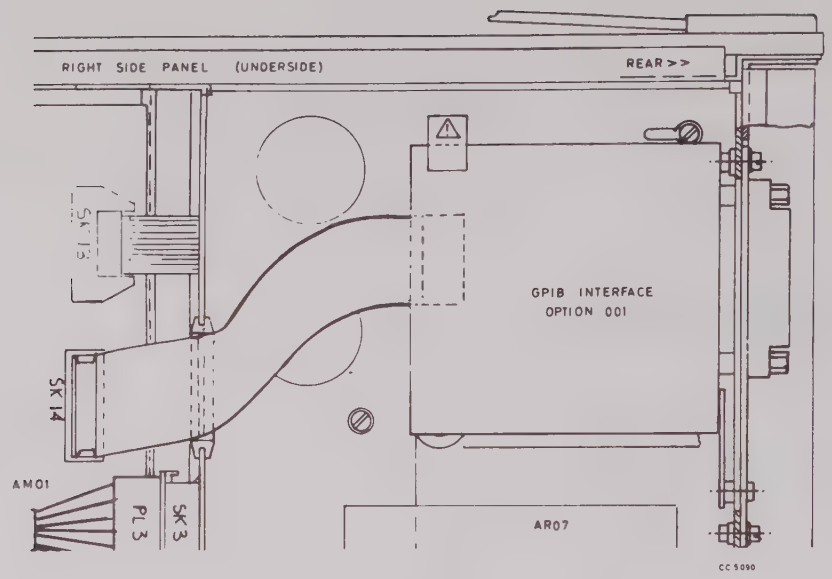



*Fig. 1 Interface installation*

(4) Add the two supplied self-adhesive labels to the rear panel identifying the GPIB connector and address switch as shown in Fig. 2.

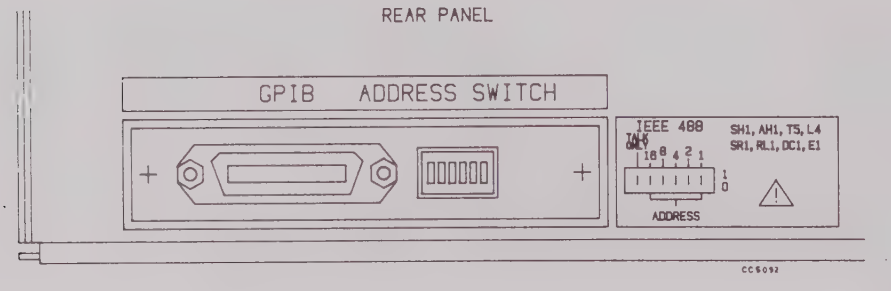



*Fig. 2 Label positioning*

(5) Replace the instrument's bottom cover.

The instrument is now ready for GPIB operation.

## GPIB ADDRESS/TALK ONLY SELECTION

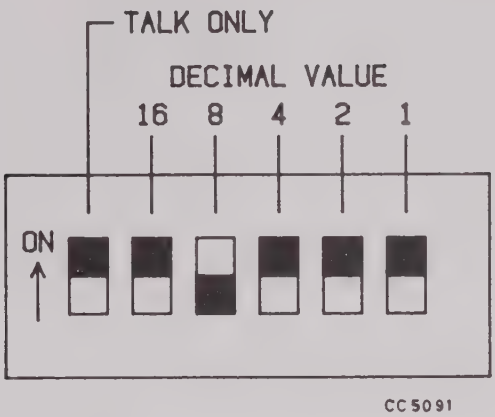



*Fig. 3 Address switch configuration*

2. The GPIB address switch is shown in Fig. 3. For normal operation the TALK ONLY switch should be off and the required address should be selected on the five address switch positions using the binary weighted decimal values shown. The interface is supplied pre-set to address 8. The normal valid address range is 0 to 30. Before changing the address check your controller manuals for reserved addresses.

3. TALK ONLY mode is activated by placing the TALK ONLY switch in the ON position. This mode is intended for use in a stand-alone configuration with a digital plotter. TALK ONLY must NOT be selected when the instrument is connected to a GPIB system governed by a controller.

4. Any change made to the GPIB address switch will only be recognized by 6500 after a power-on sequence.

Chapter 3

# OPERATION

## CONTENTS

Para.
1 Remote operation
3 GPIB program codes
5 Data entry and terminators
9 Reading data from 6500
11 Command description format
13 Measurement mode selection
14 Memory operation
15 Amplitude scaling
16 Frequency scaling
17 Sweep mode selection
18 Display mode selection
19 Limit checking facilities
20 X-Y plotter control
21 Digital plotter control
22 Instrument settings control
23 Brightline positioning
24 Marker control
25 Brightline data transfer
26 Reading status information
27 Reading display parameters
28 ASCII measurement transfer
29 Binary measurement transfer
30 Text display control
31 Keyboard control
32 SRQ facilities
33 Synthesizer mode
34 Miscellaneous commands

## Remote operation

1. On receipt of the Remote Enable (REN) signal from the GPIB the 6500 switches to remote operation, indicated by the LOCAL lamp being extinguished. No instrument settings are changed but all front panel keys except LOCAL are disabled and their functions come under GPIB control.

2. Unless inhibited by the Local Lockout (LLO) message the instrument can be returned to LOCAL control by pressing the LOCAL key or sending the Go To Local (GTL) or Interface Clear (IFC) messages or releasing the Remote Enable (REN) signal.

GPIB program codes

3. All valid GPIB program codes are listed in Table 1 and Table 2. Table 1 lists those codes which correspond directly to front panel key strokes, while Table 2 lists all others. Note that access to the secondary key functions is achieved by using the primary function code prefixed with 'SH'. 'SH' is treated by the instrument as a program code corresponding to the SHIFT key.

4. Program codes may be combined in a string without individual terminators, although spaces or commas may be included to improve clarity. These will be ignored by the instrument.

*TABLE 1*

*PRIMARY KEY-STROKE COMMANDS*

| | |
|---|---|
| CA/B/R/C | Select channel A/B/R/A&B |
| SLR | Subtract Live Reference (-R) |
| STA/B/R/C | Store to memory A/B/R/A&B |
| SBA/B/R | Subtract memory A/B/R |
| U | Change measurement units |
| SD<n>E | Set screen datum dB(m) |
| SR<n>E | Set screen range dB(m)/division |
| SS<n>E | Set start frequency GHz |
| SP<n>E | Set stop frequency GHz |
| E | Enter (follows numeric data) |
| AR | Auto range |
| FW | Full width sweep (F1-F2) |
| DF<n>E | Sweep about brightline GHz |
| MX | Set brightline to max point |
| MN | Set brightline to min point |
| MK | Marker place/remove |
| N | Normal sweep mode |
| AV | Sweep averaging mode |
| FZ | Sweep freeze mode |
| H | Select histogram display |
| L | Select line display |
| P | Enter plot mode |

*SECONDARY KEY-STROKE COMMANDS (SH = SHIFT)*

| | |
|---|---|
| SHCA/B/R | Display memory A/B/R |
| SHSTA/B/R/C | Store average to mem A/B/R/A&B |
| SHSD<n>E | Set high limit dB(m) |
| SHSR<n>E | Set low limit dB(m) |
| SHSS<n>E | Set sweeper minimum (F1) GHz |
| SHSP<n>E | Set sweeper maximum (F2) GHz |
| SH1 | Calibration aid |
| SH4 | Display instrument status |
| SH5n | Store instrument settings |
| SH6n | Recall instrument settings |
| SH7 | Enable/disable limit checking |
| SH8<n>E | Set dB relative (A) |
| SH9<n>E | Set dB relative (B) |
| SH- | Secret (blank frequency) mode |
| SHAR | Zero detectors |
| SHFWnnnE | Select detector types |
| SHMK | Remove all markers |
| SHNn | Set sweep speed |

*TABLE 2*

*BLOCK TRANSFER COMMANDS*

| | |
|---|---|
| RMA/B/R | Read ASCII from channel A/B/R |
| SHRMA/B/R | Read ASCII from memory A/B/R |
| WMA/B/R<a> | Write ASCII to channel A/B/R |
| SHWMA/B/R<a> | Write ASCII to memory A/B/R |
| RYA/B/R | Read BINARY from channel A/B/R |
| SHRYA/B/R | Read BINARY from memory A/B/R |
| WYA/B/R<b> | Write BINARY to channel A/B/R |
| SHWYA/B/R<b> | Write BINARY to memory A/B/R |

*BRIGHTLINE CONTROL/DATA TRANSFER COMMANDS*

| | |
|---|---|
| BR | Move brightline right |
| BL | Move brightline left |
| BP<i>E | Set brightline position |
| RP | Read brightline position |
| RA/B/R | Read brightline amplitude A/B/R |
| RF | Read brightline frequency |
| WD<n>E | Write brightline amplitude data |

*STATUS OUTPUT COMMANDS*

| | |
|---|---|
| RS | Read status string |
| RX | Read extended status string |
| RDA/B/R | Read display parameters A/B/R |
| RDF | Read frequency parameters |
| RMK | Read marker positions |

*DIGITAL PLOTTER COMMANDS*

| | |
|---|---|
| DPA/B | Read HPGL string for trace A/B |
| DPG | Read HPGL string for graticule |
| DPL | Read HPGL string for labels |

*TEXT/KEYBOARD CONTROL COMMANDS*

| | |
|---|---|
| WTtitle$ | Write title to top line of CRT |
| TX"text" | Write text to CRT |
| DA | Display alphanumerics only |
| RK | Read code of last key pressed |
| XK | Execute last key read by RK |

*MISCELLANEOUS COMMANDS*

| | |
|---|---|
| SQnnnnn | Set SRQ mask |
| SH2 | Enter synthesizer mode |
| SH3 | Step in synthesizer mode |
| RIn | Read instrument settings store |
| WIn | Write instrument settings store |

*DATA FORMAT CONVENTIONS USED ABOVE*

| | |
|---|---|
| n | Fixed field digit |
| <n> | Floating field data (NR2) |
| <i> | Floating field integer |
| <a> | ASCII block data (NR2 or NR3) |
| <b> | BINARY block data |
| "text" | Any number of valid ASCII bytes contained within quotes. Values 0-15 are used as control codes. |

The E command terminates <n> and <i> data.

Data entry and terminators

5. Some program codes are acted upon immediately, while others require the addition of arguments to quantify their meaning. These codes fall into two groups; those requiring numeric data correctly terminated, and those requiring a single character argument with no terminator. Block transfers are a special case and are discussed separately.

6. Numeric data should conform to NR2 format as defined in IEEE 728-1982 and should be terminated with 'E'. This is treated by the 6500 as a program code for the ENTER key. This technique ensures complete compatibility with manual operation. The allowable range and resolution of the numeric data depends on the type of operation being performed.

7. Program codes requiring the 'E' terminator are :

| | |
|---|---|
| SD | SET DATUM |
| SR | SET RANGE |
| SHSD | SET HIGH LIMIT |
| SHSR | SET LOW LIMIT |
| SS | SET START FREQUENCY |
| SP | SET STOP FREQUENCY |
| SHSS | SET SWEEPER MINIMUM |
| SHSP | SET SWEEPER MAXIMUM |
| DF | SET DELTA F SWEEP WIDTH |
| SH8 | SET dB REL A |
| SH9 | SET dB REL B |
| *SHFW | SET DETECTOR TYPES |
| *BP | SET BRIGHTLINE POSITION |
| WD | WRITE DATA |

* NR2 format is not used for these commands.

8. Program codes which require a single character argument are :

| | |
|---|---|
| SH4 | STORE INSTRUMENT SETTINGS |
| SH5 | RECALL INSTRUMENT SETTINGS |
| SHN | SET SWEEP SPEED |

Note...

The SQ command has a 5 digit argument which does not require the 'E' terminator.

Reading data from the 6500

9. When addressed to talk the 6500 can output data relating to various operating parameters. Some program codes are used to instruct the instrument which data is required. The data will be output by the 6500 when next addressed to talk. The program codes used to select output data are :

| | |
|---|---|
| RA/B/R | READ BRIGHTLINE AMPLITUDE A/B/R |
| RF | READ BRIGHTLINE FREQUENCY |
| RS | READ STATUS |
| RX | READ EXTENDED STATUS |
| RMK | READ MARKER POSITIONS |
| RK | READ KEY CODE |
| RP | READ BRIGHTLINE POSITION |
| RDA/B/R | READ AMPLITUDE DISPLAY PARAMETERS A/B/R |
| RDF | READ FREQUENCY DISPLAY PARAMETERS |
| RMA/B/R | READ ASCII MEASUREMENT CHANNEL A/B/R |
| SHRMA/B/R | READ ASCII MEASUREMENT MEMORY A/B/R |
| RYA/B/R | READ BINARY MEASUREMENT CHANNEL A/B/R |
| SHRYA/B/R | READ BINARY MEASUREMENT MEMORY A/B/R |
| DPA/B | READ DIGITAL PLOT STRING TRACE A/B |
| DPG | READ DIGITAL PLOT STRING GRATICULE |
| DPL | READ DIGITAL PLOT STRING LABELS |
| RI | READ INSTRUMENT SETTINGS STORE |

10. One of the above commands must be sent before addressing 6500 to talk. The format of the data output depends on the command sent. Some outputs may be read as numeric or string data, others may only be read as string data. In some cases the data is terminated with a CR LF sequence, in others it is terminated with the EOI signal. See individual command descriptions for more detailed information.

## Command description format

11. For the purposes of detailed description the commands are divided into functional groups. In each case the detailed description consists of the following parts :

TITLE OF FUNCTIONAL GROUP
LIST OF COMMANDS DESCRIBED (with corresponding keys)
DEFINITION OF COMMAND SYNTAX (with examples)
GENERAL DESCRIPTION OF COMMAND(S)
REFERENCE TO RELATED SECTIONS (where appropriate)

The following conventions are used :

<..> Indicates a compulsory argument
[..] Indicates an optional argument or a front panel key legend

12. Application examples are given for HP Series 200 and, where appropriate, HP85 controllers in Chapter 4. The HP Series 200 range consists at present of 9816, 9826, 9836 and 9920 models and the examples given will run on any of these machines. The examples are given in the BASIC language and do not require any Series 200 BASIC extensions or HP85 enhancement ROMs other than the I/O ROM which is necessary for GPIB operation and the Advanced Programming ROM which allows the examples to be presented as sub-programs.

## Measurement mode selection

13.

| *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|
| CA CB CR CC | [A] [B] [R] [A&B] |
| SLR U | [-R] [UNITS] |
| SHCA SHCB SHCR | [MEM A] [MEM B] [MEM R] |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION*

These commands define the fundamental operating mode of the instrument and have the following individual functions :

| | |
|---|---|
| CA | Select absolute power measurement on channel A |
| CB | Select absolute power measurement on channel B |
| CR | Select absolute power measurement on channel R |
| CC | Select absolute power measurement on channels A&B |
| SLR | Subtract live reference (-R) |
| U | Change measurement units - toggles between dBm/mW and dB/VSWR |
| SHCA | Display memory A |
| SHCB | Display memory B |
| SHCR | Display memory R |

*RELATED SECTIONS*

MEMORY OPERATION (14)
READING STATUS INFORMATION (26)

## Memory operation

14.

| *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|
| STA STB STR STC | [STORE] |
| SHSTA SHSTB SHSTR SHSTC | [STORE AV] |
| SBA SBB SBR | [SUB MEM] |

*COMMAND SYNTAX*

The normal key-stroke arguments corresponding to these commands are built into the commands themselves (i.e. STA is equivalent to the key sequence [STORE] [A]). These commands are the only exception to the general key-stroke compatibility policy.

*COMMAND DESCRIPTION*

| | |
|---|---|
| STA | Store trace to memory A |
| STB | Store trace to memory B |
| STR | Store trace to memory R |
| STC | Store traces to memories A and B |
| SHSTA | Store average to memory A |
| SHSTB | Store average to memory B |
| SHSTR | Store average to memory R |
| SHSTC | Store averages to memories A and B |
| SBA | Subtract memory A |
| SBB | Subtract memory B |
| SBR | Subtract memory R |

The validity of memory operations depends on the current measurement mode. Vol. 1 of the operating manual gives a table of valid memory operations. This table is reproduced below showing valid memory commands corresponding to each measurement mode.

| *CHANNEL* | *A* | *B* | *A&B* | *R* |
|---|---|---|---|---|
| Valid store commands | [SH] STA<br>[SH] STB<br>[SH] STR | [SH] STA<br>[SH] STB<br>[SH] STR | [SH] STC | [SH] STA<br>[SH] STB<br>[SH] STR |
| Valid subtract commands | SBA<br>SBR | SBB<br>SBR | SBA<br>SBB<br>SBR | |

The STORE AV and SUB MEM commands assume that valid data exists in the memory being accessed.

*RELATED SECTIONS*

MEASUREMENT MODE SELECTION (13)
READING STATUS INFORMATION (26)

Amplitude scaling

15. *COMMANDS DESCRIBED* / *CORRESPONDING KEYS*

| | |
|---|---|
| SD SR | [DATUM] [RANGE] |
| AR | [AUTO] |
| SH8 SH9 | [dB REL A] [dB REL B] |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| SD<DATUM VALUE>E | SD-12.5E |
| SR<RANGE VALUE>E | SR1.5E |
| AR requires no argument | |
| SH8 [dB REL A OFFSET] E | SH820E |
| SH9 [dB REL B OFFSET] E | SH9-35.5E |

*COMMAND DESCRIPTION*

SD Set Datum (Range: -99.9 to +99.9dB(m) Res: 0.1dB)
SR Set Range (Range: +0.1 to +10.9dB(m) Res: 0.1dB)
AR Auto Range (selects optimum DATUM and RANGE values)
SH8 Set dB REL A (Range: -99.9 to +99.9dB Res: 0.01dB)
SH9 Set dB REL B (Range: -99.9 to +99.9dB Res: 0.01dB)

The dB REL commands do not actually alter the amplitude scaling, but the specified offset is added to the measurement data. If no argument is supplied (e.g. SH8E) then the inverse of the current brightline reading is used. This allows the current brightline point to be used as a 0 dB reference without having to enter the required value.

*RELATED SECTIONS*

READING DISPLAY PARAMETERS (27)

Frequency scaling

16. *COMMANDS DESCRIBED* / *CORRESPONDING KEYS*

| | |
|---|---|
| SS SP | [START] [STOP] |
| SHSS SHSP | [F1] [F2] |
| FW | [F1-F2] |
| DF | [DELTA F] |
| SH- | [SECRET] |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| SS [START FREQUENCY] E | SS2.05E |
| SP [STOP FREQUENCY] E | SP12.45E |
| FW Requires no argument | |
| DF [SWEEP WIDTH] E | DF1.5E |
| SH- Requires no argument | |

*COMMAND DESCRIPTION*

SS Set Start Frequency (Range: F1 to STOP Res: 0.01GHz)
SP Set Stop Frequency (Range: START to F2 Res: 0.01GHz)
SHSS Set Sweeper F1 (Range: 0 to +126GHz Res: 0.01GHz)
SHSP Set Sweeper F2 (Range: 0 to +126GHz Res: 0.01GHz)
FW Set START and STOP to current F1 and F2 values
DF Set symmetrical swbep about brightline frequency (Sweep Width Range: F2-F1 Res: 0.01GHz)
SH- Toggle SECRET mode

F1, F2, START and STOP frequency values must always satisfy the following convention:

F1 <= START <= STOP <= F2

If no argument is specified for F1 or F2 no action is taken but the current setting is echoed on the CRT. If no argument is specified for START or STOP the current brightline frequency is used. When the FW command is sent the current START and STOP settings are lost if they are different to F1 and F2.

The SH- command toggles in and out of SECRET mode. In SECRET mode the frequency scaling information is removed from the CRT.

*RELATED SECTIONS*

READING DISPLAY PARAMETERS (27)

Sweep mode selection

17.

| *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|
| N | [NORMAL] |
| AV | [AVERAGE] |
| FZ | [FREEZE] |
| SHN | [SPEED] |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| N Requires no argument | |
| AV Requires no argument | |
| FZ Requires no argument | |
| SHN<SWEEP SPEED NUMBER> | SHN5 |

*COMMAND DESCRIPTION*

N Select NORMAL sweep operation
AV Select AVERAGE sweep mode
FZ Toggle FREEZE mode
SHN Select sweep speed (Range: 0 to 9 as SPEED menu)

The N command is a general command for returning to NORMAL sweep operation. In addition to its use as an exit from AVERAGE and FREEZE modes, NORMAL can be used to abort CALAID, PLOT and STATUS modes, and any prompted command. On receipt of the N command, the 6500 switches off AVERAGE or FREEZE mode if applicable, resets the CRT display and re-starts the sweep.

The AV command selects sweep averaging (see Vol. 1 for details). Averaging is re-started each time the command is sent.

The FZ command toggles between the current sweep mode and FREEZE mode. If the current sweep mode is AVERAGE the averaging process is not re-started on toggling out of FREEZE.

The argument for the SHN command is a single digit in the range 0 to 9 corresponding to the sweep speed menu numbers. These allow sweep speeds from 70 ms (0) to 20 s (9). The power-on value is 100 ms (1). Note that the 70 ms sweep mode incurs a 50% reduction in measurement resolution.

*RELATED SECTIONS*

READING STATUS INFORMATION (26)

Display mode selection

| 18. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | H | [HIST] |
| | L | [LINE] |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION*

H Select HISTOGRAM display mode
L Select LINE display mode

The H and L commands operate on the currently displayed channel(s). The display mode for any channel will only be affected when an H or L command is executed while that channel is displayed. The display modes for each channel are undefined at power on.

*RELATED SECTIONS*

MEASUREMENT MODE SELECTION (13)

Limit checking facilities

| 19. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | SHSD SHSR | [HIGH] [LOW] |
| | SH7 | [LIMITS] |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| SHSD<HIGH LIMIT VALUE>E | SHSD3.72E |
| SHSR<LOW LIMIT VALUE>E | SHSR-30E |
| SH7 Requires no argument | |

*COMMAND DESCRIPTION*

SHSD Set High Limit (-99.99 to +99.99 dB(m) Res: 0.01dB)
SHSR Set Low Limit (-99.99 to +99.99 dB(m) Res: 0.01dB)
SHS7 Enable/Disable limit checking

Although limit values are always entered in dB(m) limit checking may still be performed in mW or VSWR modes when the limit values will be assumed to be in mW or VSWR.

Limit checking is enabled and disabled with the SH7 command which has a toggle action. The 'enabled' condition is indicated by the letter 'L' displayed in the top left corner of the CRT, immediately above the amplitude units indicator.

Limit checking is performed only on the channel(s) currently displayed. When a limit is exceeded an appropriate warning message is displayed on the top line of the CRT. The 6500 may also be programmed to issue an SRQ on this condition.

*RELATED SECTIONS*

READING DISPLAY PARAMETERS (27)
READING STATUS INFORMATION (26)
SRQ FACILITIES (32)

## X-Y Plotter control

20. *COMMANDS DESCRIBED* — *CORRESPONDING KEYS*

P — [PLOT]

*COMMAND SYNTAX*

The P command requires no argument, but invokes a menu driven plot sequence which is described below.

*COMMAND DESCRIPTION*

Assuming the 6500 is not in 'TALK ONLY' mode (see DIGITAL PLOTTER CONTROL) the P command generates an analog plotter control menu as follows:

0 - Plot All
1 - Pen Bottom Left
2 - Pen Top Right
3 - Pen to Origin
4 - Draw Axes
5 - Label Axes
6 - Plot
7 - Set Pen Lift
8 - Set Plot Speed
9 - Set Live Y

Option 0 causes options 4, 5 and 6 to be performed in sequence, producing a plot with labelled axes. Options 1, 2 and 3 are used to set up the plotter input scaling. Option 7 is used to set the logical sense of the pen lift output. A secondary menu is generated with the following options :

1 - Normal
2 - Invert

Option 8 of the primary menu allows selection of the plot speed. This is useful when using plotters which suffer from 'overshoot' problems. A plot speed number from 1 to 9 must be entered. The currently selected speed is indicated.

Option 9 allows the user to set up the plotter Y output to drive a chart recorder. A secondary menu is generated :

1 - Off
2 - On

All option numbers are sent over the bus as single ASCII digits. As some of the procedures require some time for execution two methods of ascertaining when the 6500 is waiting for an option number entry are provided. One is an SRQ facility and the other is part of the extended status message.

*RELATED SECTIONS*

DIGITAL PLOTTER CONTROL (21)
READING STATUS INFORMATION (26)
SRQ FACILITIES (32)

Digital plotter control

| 21. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | DPA DPB DPG DPL | - |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION*

DPA Read HPGL string for Trace A
DPB Read HPGL string for Trace B
DPG Read HPGL string for Graticule
DPL Read HPGL string for labels

The DPA, DPB, DPG and DPL commands allow the controller to read the HPGL strings associated with trace A, trace B, graticule and labels which would normally be sent directly to the plotter when in 'TALK ONLY' mode. On receipt of one of thee commands the 6500 will output the appropriate strings when next addressed to talk. The strings are terminated with CRLF and the EOI signal. The strings associated with traces are a fixed length of 3418 bytes. The graticule and label strings may vary in length from 350 to 1100 and 150 to 500 characters respectively depending on display content. The strings may be sent directly to a HPGL plotter or stored for later use. Appendix G gives scaling information to allow modifications to be made to the strings (e.g. to add a title).

*Digital plot in TALK ONLY mode*

The talk only facility allows a digital plotter to be driven by the 6500 in a "stand alone" configuration provided it is compatible with the following HPGL command subset.

| | |
|---|---|
| DF | Set default |
| SC | Scale |
| SR | Set relative character size |
| PA | Plot absolute |
| PU | Pen up |
| PD | Pen down |
| LB | Label |
| SP | Select pen* |

* Plotters with only one pen may be used although a plotter with two pens is preferable.

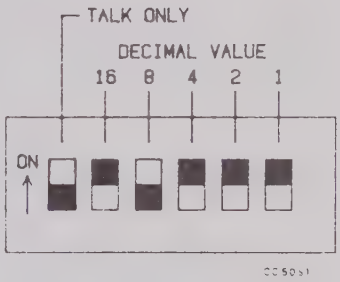
The digital plotter facilities are made available as follows:-

(1) Select the plotter to the Listen only mode (see Plotter manufacturer's Instruction Manual for details of this).

(2) Before switching 6500 power on, select the GPIB Address switch to Talk only mode.

Note...

When the 6500 is in the Talk only mode it must not be connected to a controller or to any other GPIB instrument other than a HPGL compatible plotter set to Listen only mode. This is important as a hardware bus conflict could arise. When connecting a 6500 to a controller driven system remember to check that the Talk only mode is not selected.




(3) Switch on 6500, press PLOT key which will present the following menu:

0 - Plot all
1 - Draw graticule *
2 - Label graticule **
3 - Plot
4 - Set live Y ***

* Draw graticule, a representation of the 6500 is plotted with the brightline and markers included and distinguishable from the vertical graticule lines.

** Label graticule. The following annotation is added around the graticule:

(i) Sweep speed.
(ii) Vertical scale units.
(iii) Vertical scaling.
(iv) Frequency scaling (except when SECRET mode is selected).
(v) Brightline spot measurements.
(vi) Measurement details.

Two pen colours are used to improve clarity on two channel plots.

*** Live Y. This menu option is duplicated from the analogue X Y plotter menu to allow 'Live Y output' to be selected. See Appendix H for sample of digital plot.

*RELATED SECTIONS*

X-Y PLOTTER CONTROL (20)
APPENDIX G and H

Instrument settings control

22. *COMMANDS DESCRIBED*

SH5 SH6
RI WI

*CORRESPONDING KEYS*

[STO] [RCL]
-

*COMMAND SYNTAX*

SH5<DESTINATION STORE>
SH6<SOURCE STORE>
RI<SOURCE STORE>
WI<DESTINATION STORE><BINARY DATA>

*EXAMPLE*

SH53
SH61
RI9
See APPLICATIONS (Chap. 4)

*COMMAND DESCRIPTION*

SH5 Store current instrument settings to specified store
SH6 Recall instrument settings from specified store
RI Instruct 6500 to send binary data from specified store
WI Write binary data to specified store

In all cases the source/destination store is a number in the range 1-9. SH5 causes the current instrument settings to the specified store. The following parameters are stored :

F1, F2, START, STOP
DATUM A, DATUM B, DATUM R, RANGE A, RANGE B, RANGE R
LIMIT CHECKING ON/OFF
HIGH LIMIT A, HIGH LIMIT B, LOW LIMIT A, LOW LIMIT B
dB REL A, dB REL B
SWEEP SPEED
AVERAGING ON/OFF
MARKER POSITIONS
MEASUREMENT MODE (Channels, Memory usage, Units, Etc.)
TEMPERATURE CORRECTION ON/OFF
DETECTOR TYPES

Once stored, these instrument settings can be recalled with SH6. The 6500 stores the settings internally as a binary data block. This can be read over the bus with the RI command. The data is sent as a 58 byte string conforming to the block data transfer format specified in IEEE 728-1982 as follows :

#JBBB....BBBC

Where : #J indicates a binary transfer format (IEEE 728-1982)
B is an 8 bit binary value
C is a checksum byte

The data can be sent back to any store with the WI command. The data is intended for storage/retrieval only and should not be altered. The 6500 will not accept a data transfer with an invalid checksum byte. Note that in both directions the transfer is terminated by asserting the EOI signal with the last byte.

Brightline positioning

23.

| *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|
| BR BL | ROTARY CONTROL |
| BP | - |
| MX MN | [MAX] [MIN] |
| RP | - |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| BR Requires no argument | |
| BL Requires no argument | |
| BP<BRIGHTLINE POSITION>E | BP123E |
| MX Requires no argument | |
| MN Requires no argument | |
| RP Requires no argument | |

*COMMAND DESCRIPTION*

BR Move brightline one position right
BL Move brightline one position left
BP Move brightline to specified position (Range : 0 to 421)
MX Move brightline to MAX point on displayed trace(s)
MN Move brightline to MIN point on displayed trace(s)
RP Instruct 6500 to output current brightline position

The trace display is divided into 422 brightline measurement positions and 211 physical brightline positions. Brightline control is implemented in terms of measurement position at all times. Thus multiple BL or BR commands will only move the brightline physically on alternate executions, but each execution will move the brightline to a new measurement position.

The RP command causes the 6500 to output the current brightline measurement position as a 3 digit number in the range 0 to 421 terminated with CRLF and EOI.

The 6500 can be programmed to issue an SRQ on manual operation of the brightline control when in the remote state.

*RELATED SECTIONS*

MARKER CONTROL (24)
BRIGHTLINE DATA TRANSFER (25)
SRQ FACILITIES (32)

Marker control

| 24. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | MK SHMK | MARKER ERASE |
| | RMK | - |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION*

MK Place marker at current brightline position
SHMK Erase all markers
RMK Read marker positions

A maximum of eight markers can be placed using the MK command. Markers are placed at the current brightline position and are displayed as vertical lines on the CRT similar in appearance to the brightline itself. Markers can be erased individually by placing the brightline over a displayed marker and executing MK. Alternatively, all markers can be erased with SHMK.

The RMK command causes the 6500 to output a 32 byte string consisting of eight three-digit numbers separated by commas as follows :

NNN,NNN,NNN,NNN,NNN,NNN,NNN,NNN

Each number represents the current display position of a marker in the range 0 to 210. A value of 999 indicates that the marker is not currently in use. The output string is terminated with CRLF and EOI.

*RELATED SECTIONS*

BRIGHTLINE POSITIONING (23)

Brightline data transfer

| 25. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | RA RB RR RF | - |
| | WD | - |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| RA Requires no argument | |
| RB Requires no argument | |
| RR Requires no argument | |
| RF Requires no argument | |
| WD<DATA VALUE>E | WD-30.5E |

*COMMAND DESCRIPTION*

RA Read channel A amplitude at brightline position
RB Read channel B amplitude at brightline position
RR Read channel R amplitude at brightline position
RF Read frequency at brightline position
WD Write data to displayed channel at brightline position

RA, RB and RR cause the 6500 to output the current brightline amplitude of the A, B or R channels (which do not have to be currently displayed). The

value is output in current units and has a fixed format :

SDD.DD

where S is a sign,
D is a digit 0 - 9.

RF causes the 6500 to output the current brightline frequency in GHz in the following fixed format :

SDDD.DD

The WD command allows amplitude data to be written to the currently displayed trace. The data should conform to NR2 format as defined in IEEE 728-1982 and must be terminated with 'E'. The display must be in FREEZE mode. Data is written to the current brightline position and is assumed to be in current units. When both A and B traces are displayed, the data is written to channel A only.

*RELATED SECTIONS*

BRIGHTLINE POSITIONING (23)
SWEEP MODE SELECTION (17)

## Reading status information

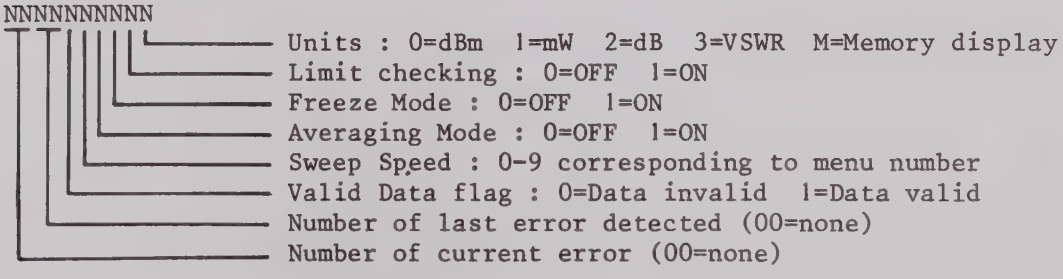
| 26. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | RS RX | - |

*COMMAND SYNTAX*

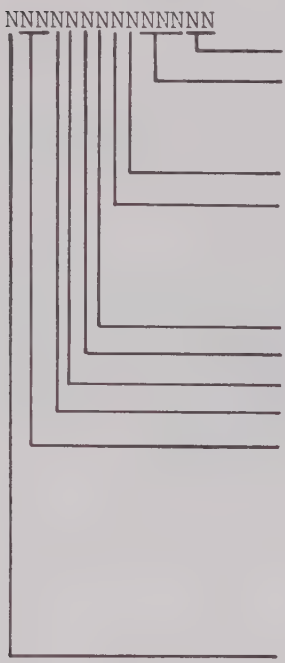
These commands do not require any arguments.

*COMMAND DESCRIPTION*

RS causes the 6500 to output a 10 character string containing status information as follows :


NNNNNNNNNN
Units : 0=dBm 1=mW 2=dB 3=VSWR M=Memory display
Limit checking : 0=OFF 1=ON
Freeze Mode : 0=OFF 1=ON
Averaging Mode : 0=OFF 1=ON
Sweep Speed : 0-9 corresponding to menu number
Valid Data flag : 0=Data invalid 1=Data valid
Number of last error detected (00=none)
Number of current error (00=none)

RX causes the output of a 24 character string consisting of the above plus 14 additional characters defined as follows :




- 6500 Firmware Issue Number
- Detector type ABR. Each digit identifies the detector type by the last digit of its type number (i.e. 1=6511 4=6514 etc.).
- Temperature Correction : 0=OFF 1=ON
- Memory status : (may be combined to give 0-7)
  - 0 - All memories null
  - 1 - Memory A valid
  - 2 - Memory B valid
  - 4 - Memory R valid
- Live Y output : 0=OFF 1=ON
- Plot Speed : 1-9 as defined in plot menu
- Pen Lift Sense : 0=Normal 1=Inverted
- Plot mode : 0=Inactive 1=Awaiting Command 2=Busy
- Measurement Mode : (may be combined to give 00-63)
  - 00 - Absolute power
  - 01 - Sub Mem A
  - 02 - Sub Mem B
  - 04 - Sub Mem R
  - 08 - -REF
  - 16 - dB REL A
  - 32 - dB REL B
- Display Mode :
  - 0 - Channel A
  - 1 - Channel B
  - 2 - Channel R
  - 3 - Channels A&B
  - 4 - MEM A
  - 5 - MEM B
  - 6 - MEM R
  - 7 - CALAID

*RELATED COMMANDS*

The use of RS and RX is discussed in APPLICATIONS (Chap. 4).

Reading display parameters

| 27. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | RDA RDB RDR RDF | - |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION*

Each of these commands causes the 6500 to output a string containing four numeric values separated by commas. The values output are as follows :

| | | | | |
|---|---|---|---|---|
| RDA | DATUM A, | RANGE A, | HIGH LIM A, | LOW LIM A |
| RDB | DATUM B, | RANGE B, | HIGH LIM B, | LOW LIM B |
| RDR | DATUM R, | RANGE R, | dB REL A, | dB REL B |
| RDF | F1 , | F2 , | F START , | F STOP |

For RDA, RDB and RDR a 27 byte string is output in the following format :

SDD.DD,SDD.DD,SDD.DD,SDD.DD

For RDF a 31 byte string is output as follows :

SDDD.DD,SDDD.DD,SDDD.DD,SDDD.DD

In each case S indicates a sign, D indicates a digit and the string is terminated with CRLF and EOI.

*RELATED SECTIONS*

AMPLITUDE SCALING (15)
FREQUENCY SCALING (16)
LIMIT CHECKING FACILITIES (19)

ASCII measurement transfer

| 28. *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|
| RMA RMB RMR SHRMA SHRMB SHRMR | - |
| WMA WMB WMR SHWMA SHWMB SHWMR | - |

*COMMAND SYNTAX*

ASCII read commands (prefix R or SHR) require no parameters . ASCII write commands (prefix W or SHW) must be followed by a block of ASCII values as described below.

*COMMAND DESCRIPTION*

| | |
|---|---|
| RMA | Read ASCII measurement from channel A |
| RMB | Read ASCII measurement from channel B |
| RMR | Read ASCII measurement from channel R |
| SHRMA | Read ASCII measurement from memory A |
| SHRMB | Read ASCII measurement from memory B |
| SHRMR | Read ASCII measurement from memory R |

In each case the data output consists of 422 values separated by commas forming a 2953 byte string as follows :

SDD.DD,SDD.DD, ..... ,SDD.DD,SDD.DD
1st value, ..... ,422nd value

Where S indicates a sign and D indicates a digit. The string is terminated with CRLF and EOI.

| | |
|---|---|
| WMA | Write ASCII measurement to channel A |
| WMB | Write ASCII measurement to channel B |
| WMR | Write ASCII measurement to channel R |
| SHWMA | Write ASCII measurement to memory A |
| SHWMB | Write ASCII measurement to memory B |
| SHWMR | Write ASCII measurement to memory R |

In each case the command must be followed by 422 data values conforming to NR2 or NR3 formats as defined in IEEE 728-1982 separated by commas and terminated with CRLF, LF only or EOI.

Note that measurements may be transferred to any channel or memory irrespective of whether it is currently being displayed. When writing to a channel the 6500 should be in FREEZE mode in order to retain the data.

*RELATED SECTIONS*

BINARY MEASUREMENT TRANSFER (29)
ASCII transfers are discussed in APPLICATIONS (Chap. 4).

## Binary measurement transfer

| 29. *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|
| RYA RYB RYR SHRYA SHRYB SHRYR | - |
| WYA WYB WYR SHWYA SHWYB SHWYR | - |

*COMMAND SYNTAX*

Binary read commands (prefix R or SHR) require no parameters. Binary write commands (prefix W or SHW) must be followed by a binary data block as described below.

*COMMAND DESCRIPTION*

| | |
|---|---|
| RYA | Read binary measurement from channel A |
| RYB | Read binary measurement from channel B |
| RYR | Read binary measurement from channel R |
| SHRYA | Read binary measurement from memory A |
| SHRYB | Read binary measurement from memory B |
| SHRYR | Read binary measurement from memory R |

In each case the data output consists of an 846 byte string formatted as follows :

#IBBBB ..... BBBB

Where #I indicates a binary transfer format which is defined in IEEE 728-1982. B is an 8 bit binary byte. There are 422 pairs of bytes which contain data values conforming to the 6500 internal 16 bit data format. The string is terminated by asserting the EOI signal with the last data byte.

| | |
|---|---|
| WYA | Write binary measurement to channel A |
| WYB | Write binary measurement to channel B |
| WYR | Write binary measurement to channel R |
| SHWYA | Write binary measurement to memory A |
| SHWYB | Write binary measurement to memory B |
| SHWYR | Write binary measurement to memory R |

In each case the command must be followed by a binary data block as described above. The 6500 will only terminate binary data input with the EOI signal which should be asserted with the last data byte. The binary transfer facility is intended for applications where no manipulation of the data is required, although a technique for extracting the measurement values is described in APPLICATIONS (Chap. 4). The main advantage of binary over ASCII transfers is an approximate 5:1 speed improvement. (The actual transfer times depend on the GPIB controller being used - see Chap. 4).

*RELATED SECTIONS*

ASCII MEASUREMENT TRANSFER (28)
Binary transfers are discussed in APPLICATIONS (Chap. 4).

Text display control

30. *COMMANDS DESCRIBED* — *CORRESPONDING KEYS*

WT TX DA — -

*COMMAND SYNTAX* — *EXAMPLE*

| | |
|---|---|
| WT<TEXT>$ | WTBand Pass Filter$ |
| TX<"TEXT"> | TX"Press [ENTER] :" |
| DA Requires no argument | |

*COMMAND DESCRIPTION*

WT is used to write a title to the top line of the CRT (overwriting the measurement mode indicator) with a maximum length of 28 characters. All printable ASCII characters may be used except '$' which is used as a terminator.

TX may be used to write text anywhere on the 6500 CRT. The 6500 maintains a 'text pointer' similar to a cursor on a VDU. Text written with TX will always start at the current text pointer position. The position of the text pointer is incremented by one column (horizontally) with each character of the text until the end of the current row (column 40) is reached. All subsequent text will overwrite the last character on the row. The text pointer may be manipulated in various ways with ASCII codes 0 to 15 which are reserved as control codes. A full list of these is given in Appendix C. Control facilities include :

Text pointer move in any direction
Clear screen/row
Set/Reset tab stops
Enable/Disable flashing characters

All standard ASCII characters plus 16 special characters are available. A table showing these is given in Appendix C. There is no limit to the length of the text but in order to comply with the text transfer format defined in IEEE 728-1982 it must be contained within quotes (").

The DA command switches off all display elements except alphanumerics. Thus text can be written over existing annotation or the CRT can be used simply as a VDU. Appendix E contains a blank CRT layout chart (the CRT format is 24 rows of 40 columns) and Appendix F contains layout charts showing the positions of normal 6500 annotation and messages. Normal display mode may be resumed with the N (NORMAL) command.

*RELATED SECTIONS*

APPENDICES C,E,F
Text control is discussed in APPLICATIONS (Chap. 4).

Keyboard control

| 31. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | RK XK | - |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION*

RK Read last key pressed
XK Execute last key read by RK

The RK command is normally used in conjunction with the 'SRQ on key press' facility (see SRQ FACILITIES) to intercept key presses while the 6500 is under remote control. The 6500 will output a number in the range 0 to 38 indicating which key was pressed. In order to make the interpretation of numeric data entry easier the numeric keys 0 to 9 are assigned key codes 0 to 9. APPENDIX D shows the keyboard layout indicating all the key codes.

When a key code has been read with RK the XK command may be used to make the 6500 execute the procedure associated with that key as though it had been pressed under local control. Using this technique it is possible to inhibit certain keys, or add extra user-prompts and facilities to an instrument apparently operating under local control. These ideas are discussed further in APPLICATIONS (Chap. 4).

*RELATED SECTIONS*

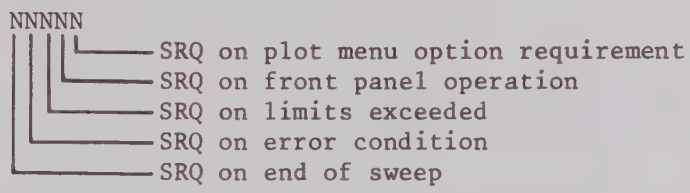
SRQ FACILITIES (32)
APPENDIX D

SRQ facilities

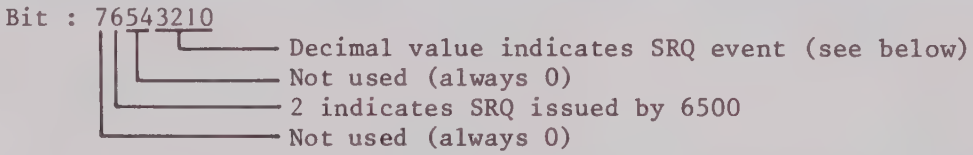
| 32. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | SQ | - |
| | *COMMAND SYNTAX* | *EXAMPLE* |
| | SQ<SRQ MASK> | SQ10000 |

*COMMAND DESCRIPTION*

The SRQ mask allows SRQ interrupts to be enabled/disabled for any combination of five events. The mask consists of a 5 digit ASCII string. Each digit should be either 0 to disable the function or 1 to enable it. The functions are arranged as follows :


NNNNN
SRQ on plot menu option requirement
SRQ on front panel operation
SRQ on limits exceeded
SRQ on error condition
SRQ on end of sweep

When one of the enabled conditions occurs, an SRQ will be issued. The controller should respond with a SERIAL POLL interrogation. The 6500 will then output a STATUS BYTE. The STATUS BYTE is interpreted as follows :


Bit : 76543210
Decimal value indicates SRQ event (see below)
Not used (always 0)
2 indicates SRQ issued by 6500
Not used (always 0)

The decimal value of bits 0-3 is interpreted as follows :

| | |
|---|---|
| 0 | Not used |
| 1 | Key press |
| 2 | Brightline control rotated left |
| 3 | Brightline control rotated right |
| 4-9 | Not used |
| 10 | End of sweep |
| 11 | Limits exceeded (only if limits enabled) |
| 12 | Error condition |
| 13 | Plot menu option requirement |
| 14-15 | Not used |

Note...

PARALLEL POLL is not supported in 6500.

*RELATED SECTIONS*

The use of SRQ facilities is discussed further in APPLICATIONS (Chap. 4).

Synthesizer mode

| 33. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | SH2 SH3 | - |

*COMMAND SYNTAX*

These commands do not require any arguments.

*COMMAND DESCRIPTION.*

The Synthesizer Mode commands are provided to allow the use of GPIB programmable frequency synthesizers where no external ramp input facility exists. Synthesizer mode is entered with the SH2 command. The 6500 now operates as normal except that it waits for the SH3 command before making a new measurement. Thus 422 SH3 commands are required to perform a complete sweep. The controller must send appropriate frequency commands to the synthesizer in between each SH3 command execution. All normal measurement modes may be used. Note that some commands (SD, SR, SS, SP, AR, etc.) cause the 6500 to re-start its sweep. The controlling software must take account of this in order to maintain correct frequency tracking between the 6500 and the synthesizer. To exit from synthesizer mode use the N command.

*RELATED SECTIONS*

An example of Synthesizer Mode operation is given in APPLICATIONS (Chap. 4).

Miscellaneous functions

| 34. | *COMMANDS DESCRIBED* | *CORRESPONDING KEYS* |
|---|---|---|
| | SHAR | [ZERO] |
| | SHFW | [DET] |
| | SH1 | [CALAID] |
| | SH4 | [STATUS] |

| *COMMAND SYNTAX* | *EXAMPLE* |
|---|---|
| SHAR Requires no argument | |
| SHFW<DETECTOR TYPES>E | SHFW114E |
| SH1 Requires no argument | |
| SH4 Requires no argument | |

*COMMAND DESCRIPTION*

SHAR performs an AUTO-ZERO operation on the detectors. No r.f. power should be connected during this operation. An AUTO-ZERO operation MUST be performed after power on before any measurements are made. Consult Vol. 1 of the Operating Manual for further details.

SHFW is used to select appropriate error correction for the detector types being used. The argument is a 3 digit number where each number identifies the detector type by the last digit of its type number (i.e. 1=6511 4=6514 etc.). The 3 digits identify detector types for channels A, B and R respectively.

SH1 and SH4 are included for the sake of completeness but are not intended for remote operation. SH1 invokes the calibration aid facility (which can be aborted with the N command), and SH4 calls up the instrument status display. Both of these facilities are essentially manual functions and the user should consult Vol. 1 of the Operating Manual for further details of their operation.

*RELATED SECTIONS*

READING STATUS INFORMATION (26)

Chapter 4

# APPLICATIONS

## CONTENTS

Para.
1 Introduction
2 HP Series 200 Implementation
3 HP85 Implementation
5 Description format of examples
6 Example 1 Valdat
7 Example 2 Freeze
8 Example 3 Autoz
9 Example 4 Synth
10 Example 5 Maxmin
11 Example 6 Blpos
12 Example 7 Mfreq
13 Example 8 Dplot
14 Example 9 Readi
15 Example 10 Writei
16 Example 11 Reada
17 Example 12 Writea
18 Example 13 Readb
19 Example 14 Writeb
20 Example 15 Writet
21 Example 16 Keysrq
22 Example 17 Plocal
27 Binary/ASCII conversions

## INTRODUCTION

1. This chapter presents a set of example GPIB program routines which are designed to cover some of the common system requirements and to illustrate some of the advanced features of the 6500 GPIB facility. All the examples are presented as sub-programs or functions which may be incorporated directly into the user's programs. For each example an implementation is given for the HP Series 200 controller and, where appropriate, the HP85. All examples are fully commented. The comments (preceded by '!') may, of course, be omitted.

## HP SERIES 200 IMPLEMENTATION

2. Currently the Series 200 range of controllers includes the following models:

9816 (216, Model 16)
9826 (226, Model 26)
9836 (236, Model 36)
9920 (220, Model 20)

The example routines will run on any of the above models with BASIC 2.0 installed. No language extensions are required. The example routines are presented as SUB-PROGRAMS or FUNCTIONS which provide a self-contained program context. The routines should be called with parameters where necessary. Parameters must be of the correct type (REAL, INTEGER or STRING). The main program should contain COM statements to match those in the sub-program and should perform variable initialization, I/O path assignments and array/string dimensioning as necessary. GPIB device addresses and status string variables are passed in COM statements. The following conventions have been adopted:

```
COM /A/ @Analyser       6500 GPIB address
COM /B/ Status$         6500 status string (24 chars.)
COM /C/ @Synthesizer    Synthesizer GPIB address
COM /D/ @Plotter        Plotter GPIB address
```

Consult the controller manuals for detailed information on programming semantics.

## HP85 IMPLEMENTATION

3. The HP85 routines are presented as SUB-PROGRAMS, which require the ADVANCED PROGRAMMING ROM to be installed. This approach was adopted as it allows the routines to be presented in a convenient manner. For use without the AP ROM all the routines may be easily converted to normal BASIC sub-routines with the following procedure :

(a) Remove the SUB definition statement.

(b) Remove COM statements or replace with DIM statements if necessary.

(c) Replace SUBEXIT and, where appropriate, SUBEND statements with RETURN.

(d) Replace CALL statements with GOSUB statements to the appropriate line number for the routine being called.

Note that variables which are local to the sub-program will become global when the above procedure is followed.

4. In addition to the AP ROM the following hardware items are required for correct operation :

```
ROM DRAWER
I/O ROM
HPIB INTERFACE
```

For use as sub-programs the routines should be called with parameters where necessary and any COM statements in the sub-program should have matching COM statements in the main program. All variable initialization should be performed in the main program. GPIB addresses and status string variables are passed in COM statements. The following conventions have been adopted :

```
COM A1          6500 GPIB address
COM S$[24]      6500 status string
COM A2          Synthesizer GPIB address
COM A3          Plotter GPIB address
```

Consult the controller manuals for full details of programming semantics.

## DESCRIPTION FORMAT OF EXAMPLES

5. In addition to full listings of each implementation a description of each example is given using the following format :

Title and general description
Input requirements (formal parameters)
Output definitions (formal parameters)
Sub-program calls
Special considerations

Example 1 Valdat

6. This routine monitors 6500 status until the valid data flag is true. Measurement data becomes invalid under certain conditions (during AUTO RANGE, AUTO ZERO, etc.) and certain operations require data to be valid before they can be performed successfully. GPIB commands requiring valid data are :

[SH] STA [SH] STB [SH] STC FZ P MX MN

It is good practice to call Valdat before executing any of these commands.

Input requirements : NONE

Output definitions : NONE

Sub-program calls : NONE

*HP85 Sub-program*

```
1000 SUB "Valdat"
1010 COM A1
1020 COM S$[24]
1030 OUTPUT A1 ;"RS" !                          Tell 6500 to output status
1040 ENTER A1 ; S$ !                            Read status string
1050 IF S$[5,5]<>"1" THEN WAIT 100 @ GOTO 1030 !  Repeat if data not valid
1060 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Valdat
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   LOOP                                    ! Checking loop
10040     OUTPUT @Analyser;"RS"                 ! Tell 6500 to output status
10050     ENTER @Analyser;Status$               ! Read status string
10060   EXIT IF Status$[5,5]="1"                ! Exit loop if data valid
10070     WAIT .1                               ! Else wait 100mS
10080   END LOOP                                ! and try again
10090 SUBEND
```

## Example 2 Freeze

7. This routine places 6500 into freeze mode. No action is taken if freeze mode is already active.

Input requirements : NONE

Output definitions : NONE

Sub-program calls : Valdat

*HP85 Sub-program*

```
1000 SUB "Freeze"
1010 COM A1
1020 COM S$[24]
1030 OUTPUT A1 ;"RS" !                          Tell 6500 to output status
1040 ENTER A1 ; S$ !                            Read status string
1050 IF S$[8,8]="1" THEN SUBEXIT !              Return if already in FREEZE mode
1060 CALL "Valdat" !                            Wait for valid data
1070 OUTPUT A1 ;"FZ" !                          Place in FREEZE mode
1080 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Freeze
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   OUTPUT @Analyser;"RS"                   ! Tell 6500 to output status
10040   ENTER @Analyser;Status$                 ! Read status string
10050   IF Status$[8,8]="1" THEN SUBEXIT        ! Return if already in FREEZE mode
10060   CALL Valdat                             ! Wait for valid data
10070   OUTPUT @Analyser;"FZ"                   ! Place in FREEZE mode
10080 SUBEND
```

Example 3 Autoz

8. This routine performs an AUTO ZERO operation and returns with a flag indicating success or failure.

Input requirements : NONE

| Output definitions : | Series 200 | HP85 |
|---|---|---|
| Error flag (1 = failure) | see below | E |

Sub-program calls : Valdat

The Series 200 version of this routine is implemented as a function and should be called by :

Variable=FNAutoz

*HP85 Sub-program*

```
1000 SUB "Autoz" (E)
1010 COM A1
1020 COM S$[24]
1030 OUTPUT A1 ;"N" !                          NORMAL clears any current errors
1040 CALL "Valdat" !                           Wait for valid data
1050 OUTPUT A1 ;"SHAR" !                       Initiate AUTO ZERO
1060 CALL "Valdat" !                           Wait for valid data
1070 IF S$[1,2]="42" THEN E=1 ELSE E=0 !       Set or clear error flag
1080 SUBEND
```

*HP200 Sub-program*

```
10000 DEF FNAutoz
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   OUTPUT @Analyser;"N"                   ! NORMAL clears any current error
10040   CALL Valdat                            ! Wait for valid data
10050   OUTPUT @Analyser;"SHAR"                ! Initiate AUTO ZERO
10060   CALL Valdat                            ! Wait for valid data
10070   IF Status$[1,2]="42" THEN              ! Check for failure
10080     Fail=1                               ! Set fail flag
10090   ELSE
10100     Fail=0                               ! or clear it
10110   END IF
10120   RETURN Fail
10130 FNEND
```

Example 4 Synth

9. This routine generates an autoranged sweep using the 6500 GPIB 'synthesizer' mode.

| Input requirements : | Series 200 | HP85 |
|---|---|---|
| Start frequency | Fstart | F1 |
| Stop frequency | Fstop | F2 |

Output definitions : NONE

Sub-program calls : NONE

Note that the AUTO RANGE function requires two sweeps, the first of which is not displayed.

In the example the synthesizer is assumed to accept frequency data in units of 10 kHz (e.g. Marconi 6812). The units conversion statement may need to be altered to suit other synthesizers.

*HP85 Sub-program*

```
1000 SUB "Synth" (F1,F2)
1010 COM A1
1020 COM A2
1030 OUTPUT A1 ;"SS"&F1&"E" !          Set start frequency
1040 OUTPUT A1 ;"SP"&F2&"E" !          Set stop frequency
1050 OUTPUT A1 ;"SH2" !                Initiate synthesizer mode
1060 OUTPUT A1 ;"AR" !                 Initiate AUTO RANGE
1070 FOR S=1 TO 2 !                    Two sweeps required for AUTO
1080 WAIT 100 !                        Allow set up
1090 FOR P=0 TO 421 !                  Sweep loop
1100 F=F1+P*(F2-F1)/421 !              Calculate frequency
1110 F=INT(F*100000+.5) !              Convert to 10kHz units
1120 OUTPUT A2 ;F !                    Output to synthesizer
1130 WAIT 10 !                         Allow synthesizer to lock
1140 OUTPUT A1 ;"SH3" !                Step 6500
1150 NEXT P !                          End of sweep loop
1160 NEXT S
1170 WAIT 100 !                        Allow processing
1180 OUTPUT A1 ;"FZ" !                 Freeze display
1190 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Synth(Fstart,Fstop)
10010   COM /A/ @Analyser
10020   COM /C/ @Synthesizer
10030   OUTPUT @Analyser;"SS";Fstart;"E"                ! Set start frequency (for display only)
10040   OUTPUT @Analyser;"SP";Fstop;"E"                 ! Set stop  frequency (for display only)
10050   OUTPUT @Analyser;"SH2"                          ! Initiate synthesizer mode
10060   OUTPUT @Analyser;"AR"                           ! Initiate AUTO RANGE
10070   FOR Sweep=1 TO 2                                ! Two sweeps required for AUTO RANGE
10080     WAIT .1                                       ! Allow set up
10090     FOR Point=0 TO 421                            ! Sweep loop
10100       Frequency=Fstart+Point*(Fstop-Fstart)/421   ! Calculate frequency
10110       Frequency=INT(Frequency*100000+.5)          ! Convert to 10kHz units
10120       OUTPUT @Synthesizer;Frequency               ! Output to synthesizer
10130       WAIT .01                                    ! Allow synthesizer to lock
10140       OUTPUT @Analyser;"SH3"                      ! Step 6500
10150     NEXT Point                                    ! End of sweep loop
10160   NEXT Sweep
10170   WAIT .1                                         ! Allow processing
10180   OUTPUT @Analyser;"FZ"                           ! Freeze display
10190 SUBEND
```

## Example 5 Maxmin

10. This routine returns the amplitude and frequency values of the brightline MAX and MIN functions. Markers are placed at the MAX and MIN positions.

Input requirements : NONE

Output definitions :

| | Series 200 | HP85 |
|---|---|---|
| MAX amplitude | Amax | V1 |
| MAX frequency | Fmax | F1 |
| MIN amplitude | Amin | V2 |
| MIN frequency | Fmin | F2 |

Sub-program calls : Valdat

*HP85 Sub-program*

```
1000 SUB "Maxmin" (V1,F1,V2,F2)
1010 COM A1
1020 COM S$[24]
1030 CALL "Valdat" !                          Wait for valid data
1040 OUTPUT A1 ;"SHMK" !                      Erase existing markers
1050 OUTPUT A1 ;"MX MK" !                     Find MAX and place marker
1060 OUTPUT A1 ;"RA" !                        Read amplitude
1070 ENTER A1 ; V1
1080 OUTPUT A1 ;"RF" !                        Read frequency
1090 ENTER A1 ; F1
1100 OUTPUT A1 ;"MN MK" !                     Find MIN and place marker
1110 OUTPUT A1 ;"RA" !                        Read amplitude
1120 ENTER A1 ; V2
1130 OUTPUT A1 ;"RF" !                        Read frequency
1140 ENTER A1 ; F2
1150 SUBEND
```

*HP 200 Sub-program*

```
10000 SUB Maxmin(Amax,Fmax,Amin,Fmin)
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   CALL Valdat                           ! Wait for valid data
10040   OUTPUT @Analyser;"SHMK"               ! Erase existing markers
10050   OUTPUT @Analyser;"MX MK"              ! Find MAX and place marker
10060   OUTPUT @Analyser;"RA"                 ! Read amplitude
10070   ENTER @Analyser;Amax
10080   OUTPUT @Analyser;"RF"                 ! Read frequency
10090   ENTER @Analyser;Fmax
10100   OUTPUT @Analyser;"MN MK"              ! Find MIN and place marker
10110   OUTPUT @Analyser;"RA"                 ! Read amplitude
10120   ENTER @Analyser;Amin
10130   OUTPUT @Analyser;"RF"                 ! Read frequency
10140   ENTER @Analyser;Fmin
10150 SUBEND
```

Example 6 Blpos

11. This routine positions the brightline at a target frequency. If the target frequency is outside the current START/STOP frequency range no action is taken and the error flag is returned true.

| | | |
|---|---|---|
| Input requirements : | Series 200 | HP85 |
| Target frequency | Target | F |
| Output definitions : | Series 200 | HP85 |
| Error flag | Err_flag | E |

Sub-program calls : NONE

*HP85 Sub-program*

```
1000 SUB "Blpos" (F,E)
1010 COM A1
1020 OUTPUT A1 ;"RDF" !                          Read start/stop frequencies
1030 ENTER A1 ; D,D,F1,F2
1040 IF F<F1 OR F>F2 THEN E=1 @ SUBEXIT !        If invalid F return with error
1050 E=0 !                                       Reset error flag
1060 P=INT((F-F1)/(F2-F1)*421+.5) !              Calculate brightline position
1070 OUTPUT A1 ;"BP";P;"E" !                     Place brightline
1080 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Blpos(Target,Err_flag)
10010   COM /A/ @Analyser
10020   OUTPUT @Analyser;"RDF"                   ! Read start/stop frequencies
10030   ENTER @Analyser;Dummy,Dummy,Fstart,Fstop
10040   IF Target<Fstart OR Target>Fstop THEN    ! Check target within limits
10050     Err_flag=1                             ! Return with flag set if error
10060     SUBEXIT
10070   ELSE
10080     Err_flag=0                             ! Reset error flag
10090   END IF
10100   Point=INT((Target-Fstart)/(Fstop-Fstart)*421+.5)   ! Calculate brightline position
10110   OUTPUT @Analyser;"BP";Point;"E"          ! Place brightline
10120 SUBEND
```

Example 7 Mfreq

12. This routine returns the frequency values for each of the eight markers. Values for non-displayed markers are set to -1.

Input requirements : NONE

| Output definitions : | Series 200 | HP85 |
|---|---|---|
| Marker frequency array | Frequency (*) | F( ) |

Sub-program calls : NONE

The frequency array should be dimensioned with 8 elements (0-7).

*HP85 Sub-program*

```
1000 SUB "Mfreq" (F())
1010 COM A1
1020 DIM P(7) !                                          Array for positions
1030 OUTPUT A1 ;"RDF" !                                  Read start/stop frequencies
1040 ENTER A1 ; D,D,F1,F2
1050 S=F2-F1 !                                           Calculate frequency span
1060 OUTPUT A1 ;"RMK" !                                  Read marker positions
1070 ENTER A1 ; P(0),P(1),P(2),P(3),P(4),P(5),P(6),P(7)
1080 FOR M=0 TO 7 !                                      For each marker
1090 IF P(M)=999 THEN F(M)=-1 ELSE F(M)=P(M)/210*S+F1 !  Calculate frequency (-1=unused)
1100 NEXT M
1110 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Mfreq(Frequency(*))
10010   COM /A/ @Analyser
10020   ALLOCATE INTEGER Pos(7)                          ! Allocate array for positions
10030   OUTPUT @Analyser;"RDF"                           ! Read start/stop frequencies
10040   ENTER @Analyser;Dummy,Dummy;Fstart,Fstop
10050   Span=Fstop-Fstart                                ! Calculate frequency span
10060   OUTPUT @Analyser;"RMK"                           ! Read marker positions
10070   ENTER @Analyser;Pos(*)
10080   FOR Marker=0 TO 7                                ! For each marker
10090     IF Pos(Marker)=999 THEN                        ! Ignore if unused
10100       Frequency(Marker)=-1
10110     ELSE
10120       Frequency(Marker)=(Pos(Marker)/210)*Span+Fstart   ! Calculate frequency
10130     END IF
10140   NEXT Marker
10150   DEALLOCATE Pos(*)                                ! Discard position array
10160 SUBEND
```

Example 8 Dplot

13. This routine causes the current measurement display to be plotted on an HPGL compatible plotter.

Input requirements : NONE

Output definitions : NONE

Sub-program calls : Valdat

Note that the HPGL strings are transferred directly from the 6500 to the plotter. This eliminates the need for large string variable allocation in the controller. If it was required to modify the HPGL strings in any way then they would have to be read into the controller.

*HP85 Sub-program*

```
1000 SUB "Dplot"
1010 COM A1
1020 COM S$[24]
1030 COM A3
1040 A=A1-700 !                                          Find 6500 base address
1050 P=A3-700 !                                          Find plotter base address
1060 CALL "Valdat" !                                     Wait for valid data
1070 OUTPUT A1 ;"RX" !                                   Read extended status
1080 ENTER A1 ; S$
1090 M=VAL(S$[11,11]) !                                  Extract display mode value
1100 C$="DPG" !                                          Plot graticule
1110 GOSUB 2000
1120 C$="DPL" !                                          Plot labels
1130 GOSUB 2000
1140 IF M<>1 AND M<>5 THEN C$="DPA" @ GOSUB 2000 !       Plot trace A if required
1150 IF M=1 OR M=3 OR M=5 THEN C$="DPB" @ GOSUB 2000 !   Plot trace B if required
1160 SUBEXIT
2000 ! ------------------------------------------------- Routine to transfer HPGL
2010 OUTPUT A1 ;C$ !                                     Send read command to 6500
2020 SEND 7 ; UNL TALK A LISTEN P !                      Set up direct transfer
2030 RESUME 7 !                                          Initiate transfer
2040 STATUS 7,2 ; S !                                    Wait for EOI
2050 IF NOT BIT(S,3) THEN 2040
2060 ABORTIO 7 !                                         Terminate transfer
2070 RETURN
2080 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Dplot
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   COM /D/ @Plotter
10040   STATUS @Analyser,3;Ana                          ! Find 6500 base address
10050   Ana=Ana-700
10060   STATUS @Plotter,3;Pltr                          ! Find plotter base address
10070   Pltr=Pltr-700
10080   CALL Valdat                                     ! Wait for valid data
10090   OUTPUT @Analyser;"RX"                           ! Read extended status
10100   ENTER @Analyser;Status$
10110   Mode=VAL(Status$[11,11])                        ! Extract display mode value
10120   Command$="DPG"                                  ! Plot graticule
10130   GOSUB Xfer
10140   Command$="DPL"                                  ! Plot labels
10150   GOSUB Xfer
10160   IF Mode<>1 AND Mode<>5 THEN                     ! Plot trace A if displayed
10170     Command$="DPA"
10180     GOSUB Xfer
10190   END IF
10200   IF Mode=1 OR Mode=3 OR Mode=5 THEN              ! Plot trace B if displayed
10210     Command$="DPB"
10220     GOSUB Xfer
10230   END IF
10240   SUBEXIT
10250 !-------------------------------------------------------!
10260 Xfer:                                               ! Routine to transfer HPGL
10270 !-------------------------------------------------------!
10280   OUTPUT @Analyser;Command$                       ! Send read command to 6500
10290   SEND 7;UNL TALK Ana LISTEN Pltr DATA            ! Initiate direct transfer
10300   REPEAT                                          ! Wait for EOI
10310     Monitor=READIO(7,23)
10320   UNTIL BIT(Monitor,3)
10330   ABORT 7                                         ! Terminate transfer
10340   RETURN
10350 SUBEND
```

Example 9 Readi

14. This routine reads a binary string representing the instrument settings from one of the nine settings stores.

| | Series 200 | HP85 |
|---|---|---|
| Input requirements : | | |
| Source settings store (1-9) | Sto | S |
| Output definitions : | Series 200 | HP85 |
| Binary instrument settings string | Set$ | I$ |
| Error flag | Err_flag | E |

Sub-program calls: Valdat

The binary string is intended for storage/retrieval purposes only. The error flag is set if the source store number is outside the range 1-9 or if the store specified contained null data. Note the use of the EOI terminator with binary data.

*HP85 Sub-program*

```
1000 SUB "Readi" (S,I$,E)
1010 COM A1
1020 COM S$[24]
1030 IF S<1 OR S>9 THEN E=1 @ SUBEXIT !                 Check validity of store number
1040 OUTPUT A1 ;"N" !                                   Clear any current error
1050 CALL "Valdat"
1060 OUTPUT A1 ;"RI"&VAL$(S) !                          Send RI command to 6500
1070 ENTER A1 USING "#%,#%K" ; I$ !                     Read settings string
1080 OUTPUT A1 ;"RS" !                                  Check for null store error
1090 ENTER A1 ; S$
1100 IF S$[1,2]="56" THEN E=1 ELSE E=0 !                Set or clear error flag
1110 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Readi(INTEGER Sto,Set$,Err_flag)
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   IF Sto<1 OR Sto>9 THEN                          ! Check validity of store number
10040     Err_flag=1
10050     SUBEXIT
10060   END IF
10070   OUTPUT @Analyser;"N"                            ! Clear any current error
10080   CALL Valdat
10090   OUTPUT @Analyser;"RI"&VAL$(Sto)                 ! Send RI command to 6500
10100   ENTER @Analyser USING "%,-K";Set$               ! Read settings string
10110   OUTPUT @Analyser;"RS"                           ! Read status
10120   ENTER @Analyser;Status$
10130   IF Status$[1,2]="56" THEN                       ! Check for null store error
10140     Err_flag=1                                    ! Set error flag
10150   ELSE
10160     Err_flag=0                                    ! or clear it
10170   END IF
10180 SUBEND
```

## Example 10 Writei

15. This routine sends a binary string (normally obtained using the Readi routine) to one of the nine instrument settings stores.

| Input requirements : | Series 200 | HP85 |
|---|---|---|
| Destination settings store | Sto | S |
| Binary settings string | Set$ | I$ |

| Output definitions : | Series 200 | HP85 |
|---|---|---|
| Error flag | Err_flag | E |

Sub-program calls : Valdat

The error flag is set if the specified store is outside the range 1-9 or if any error occurs during data transfer (e.g. checksum error). Note the use of the EOI terminator with binary data.

*HP85 Sub-program*

```
1000 SUB "Writei" (S,I$,E)
1010 COM A1
1020 COM S$[24]
1030 IF S<1 OR S>9 THEN E=1 @ SUBEXIT !                    Check validity of store number
1040 OUTPUT A1 ;"N" !                                      Clear any current error
1050 CALL "Valdat"
1060 CONTROL 7,16 ; 129,NUM(I$[58,58]) !                   Set up EOL for EOI on last byte
1070 OUTPUT A1 USING "K" ; "WI"&VAL$(S)&I$[1,57] !         Send settings string
1080 CONTROL 7,16 ; 2,13,10 !                              Restore EOL setting
1090 OUTPUT A1 !                                           Ensure termination if error
1100 OUTPUT A1 ;"RS" !                                     Check for error condition
1110 ENTER A1 ; S$
1120 IF S$[1,2]<>"00" THEN E=1 ELSE E=0 !                  Set or clear error flag
1130 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Writei(INTEGER Sto,Set$,Err_flag)
10010   COM /A/ @Analyser
10020   COM /B/ Status$
10030   IF Sto<1 OR Sto>9 THEN                          ! Check validity of store number
10040     Err_flag=1
10050     SUBEXIT
10060   END IF
10070   OUTPUT @Analyser;"N"                            ! Clear any current error
10080   CALL Valdat
10090   OUTPUT @Analyser USING "#,K";"WI"&VAL$(Sto)&Set$ END   ! Send settings string
10100   OUTPUT @Analyser                                ! Ensure terminator following error
10110   OUTPUT @Analyser;"RS"                           ! Read status
10120   ENTER @Analyser;Status$
10130   SELECT Status$[1,2]
10140   CASE "52","53","57","58"                        ! Check for relevant errors
10150     Err_flag=1                                    ! Set error flag
10160   CASE ELSE
10170     Err_flag=0                                    ! or clear it
10180   END SELECT
10190 SUBEND
```

Example 11 Reada

16. This routine reads a measurement trace from 6500 using ASCII block transfer into a numeric array.

| Input requirements : | Series 200 | HP85 |
|---|---|---|
| ASCII read command string | Command$ | C$ |
| **Output definitions :** | **Series 200** | **HP85** |
| Measurement data array | Array (*) | A( ) |

Sub-program calls : NONE

The command string may be any valid 6500 ASCII block read command (RMA, SHRMB, etc.). The measurement data array should be dimensioned to 422 elements (0-421).

Note that the HP85 is unable to read ASCII block data directly into a numeric array and so a loop is used to enter each element with an appropriate IMAGE statement. This restraint significantly affects transfer times.

*HP85 Sub-program*

```
1000 SUB "Reada" (C$,A())
1010 COM A1
1020 OUTPUT A1 ;C$ !                          Send command to 6500
1030 FOR I=0 TO 421 !                         Enter array with loop
1040 ENTER A1 USING "#,SDD.DD,X" ; A(I)
1050 NEXT I
1060 ENTER A1 !                               Complete termination of string
1070 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Reada(Command$,Array(*))
10010   COM /A/ @Analyser
10020   OUTPUT @Analyser;Command$             ! Send command to 6500
10030   ENTER @Analyser;Array(*)              ! Read ASCII data into array
10040 SUBEND
```

## Example 12 Writea

17. This routine sends measurement data from a numeric array to 6500 using an ASCII block transfer.

| Input requirements : | Series 200 | HP85 |
|---|---|---|
| ASCII block write command string | Command$ | C$ |
| Measurement data array | Array (*) | A( ) |

| Output definitions : | Series 200 | HP85 |
|---|---|---|
| Error flag | Err_flag | E |

Sub-program calls : NONE

The command string may be any valid 6500 ASCII block write command (WMA, SHWMB, etc.). The array should contain 422 data values (array base 0). The error flag is set if any error occurs during data transfer (e.g. ASCII data overflow).

As with the Reada routine the HP85 implementation requires the use of a program loop to transfer the array elements which significantly increases transfer time.

*HP85 Sub-program*

```
1000 SUB "Writea" (C$,A(),E)
1010 COM A1
1020 CONTROL 7,16 ; 1,NUM(",") !                    Set up EOL for ',' terminator
1030 OUTPUT A1 USING "K" ; C$&VAL$(A(0)) !          Send command and first value
1040 FOR I=1 TO 420 !                               Output values 1-420 with loop
1050 OUTPUT A1 USING "K" ; A(I)
1060 NEXT I
1070 CONTROL 7,16 ; 2,13,10 !                       Restore EOL setting
1080 OUTPUT A1 USING "K" ; A(421) !                 Output last value
1090 OUTPUT A1 !                                    Ensure termination if error
1100 OUTPUT A1 ;"RS" !                              Check for error condition
1110 ENTER A1 ; S$
1120 IF S$[1,2]<>"00" THEN E=1 ELSE E=0 !           Set or clear error flag
1130 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Writea(Command$,Array(*),Err_flag)
10010   COM /A/ @Analyser
10020   ALLOCATE Status$[10]                        ! Temporary string for status
10030   OUTPUT @Analyser;Command$;Array(*)          ! Output command and array
10040   OUTPUT @Analyser                            ! Ensure termination if error
10050   OUTPUT @Analyser;"RS"                       ! Read status
10060   ENTER @Analyser;Status$
10070   SELECT Status$[1,2]
10080   CASE "50","51","52","53","54"               ! Check for relevant errors
10090     Err_flag=1                                ! Set error flag
10100   CASE ELSE
10110     Err_flag=0                                ! or clear it
10120   END SELECT
10130   DEALLOCATE Status$                          ! Discard status string
10140 SUBEND
```

Example 13 Readb

18. This routine reads measurement data into a string using binary block transfer.

| | Series 200 | HP85 |
|---|---|---|
| Input requirements : | | |
| Binary block read command string | Command$ | C$ |
| Output definitions : | Series 200 | HP85 |
| Binary measurement string | Bdata$ | B$ |

Sub-program calls : NONE

The command string may be any valid 6500 binary block read command (RYA, SHRYB, etc.). The binary data string should be dimensioned to 846 characters. Note the use of the EOI terminator with binary data.

*HP85 Sub-program*

```
1000 SUB "Readb" (C$,B$)
1010 COM A1
1020 OUTPUT A1 ;C$ !                              Send command to 6500
1030 ENTER A1 USING "@%,@%K" ; B$ !               Read BINARY data into string
1040 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Readb(Command$,Bdata$)
10010   COM /A/ @Analyser
10020   OUTPUT @Analyser;Command$                 ! Send command to 6500
10030   ENTER @Analyser USING "%,-K";Bdata$       ! Read BINARY data into string
10040 SUBEND
```

## Example 14 Writeb

19. This routine sends binary measurement data (usually acquired with the Readb routine) to 6500.

| Input requirements : | Series 200 | HP85 |
|---|---|---|
| Binary block write command string | Command$ | C$ |
| Binary measurement data string | Bdata$ | B$ |

| Output definitions : | Series 200 | HP85 |
|---|---|---|
| Error flag | Err_flag | E |

Sub-program calls : NONE

The command string may be any valid 6500 binary block write command (WYA, SHWYB, etc.). The measurement data string should contain 16-bit binary data for 422 measurements preceded by the correct block format header. The error flag is set if any error occurs during data transfer. Note the use of the EOI terminator with binary data.

*HP85 Sub-program*

```
1000 SUB "Writeb" (C$,B$,E)
1010 COM A1
1020 CONTROL 7,16 ; 129,NUM(B$[846,846]) !          Set up EOL for EOI on last byte
1030 OUTPUT A1 USING "K" ; C$&B$[1,845] !           Send command and data
1040 CONTROL 7,16 ; 2,13,10 !                       Restore EOL setting
1050 OUTPUT A1 !                                    Ensure termination if error
1060 OUTPUT A1 ;"RS" !                              Check for error condition
1070 ENTER A1 ; S$
1080 IF S$[1,2]<>"00" THEN E=1 ELSE E=0 !           Set or clear error flag
1090 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Writeb(Command$,Bdata$,Err_flag)
10010   COM /A/ @Analyser
10020   ALLOCATE Status$[10]                        ! Temporary string for status
10030   OUTPUT @Analyser USING "#,K";Command$&Bdata$ END   ! Output command and data
10040   OUTPUT @Analyser                            ! Ensure termination if error
10050   OUTPUT @Analyser;"RS"                       ! Read status
10060   ENTER @Analyser;Status$
10070   SELECT Status$[1,2]
10080   CASE "51","52","53"                         ! Check for relevant errors
10090     Err_flag=1                                ! Set error flag
10100   CASE ELSE
10110     Err_flag=0                                ! or clear it
10120   END SELECT
10130   DEALLOCATE Status$                          ! Discard status string
10140 SUBEND
```

## Example 15 Writet

20. This routine uses the TX command to position text at any location on the 6500 CRT.

| Input requirements : | Series 200 | HP85 |
|---|---|---|
| X text start location | Tabx | X |
| Y text start location | Taby | Y |
| Text string | Text$ | T$ |

Output definitions : NONE

Sub-program calls : NONE

The X location should be in the range 1-40 and the Y location should be in the range 1-24. The maximum text length will depend on the X start location but should not be more than 40 printable characters. The routine does not switch off any other display elements. This can be performed by the main program with the DA command if required.

*HP85 Sub-program*

```
1000 SUB "Writet" (X,Y,T$)
1010 COM A1
1020 DIM S$[80] !                          String for compilation
1030 INTEGER X1,Y1 !                       Position counters
1040 S$="TX"&CHR$(34)&CHR$(1) !            Set up command with TP home code
1050 Y1=Y !                                Add TP down codes
1060 IF Y1=1 THEN 1100
1070 S$=S$&CHR$(10)
1080 Y1=Y1-1
1090 GOTO 1060
1100 X1=X !                                Add TP right codes
1110 IF X1=1 THEN 1150
1120 S$=S$&CHR$(7)
1130 X1=X1-1
1140 GOTO 1110
1150 S$=S$&T$&CHR$(34) !                   Add text and close quotes
1160 OUTPUT A1 ;S$ !                       Send complete command to 6500
1170 SUBEND
```

*HP200 Sub-program*

```
10000 SUB Writet(INTEGER Tabx,Taby,Text$)
10010   COM /A/ @Analyser
10020   ALLOCATE String$[80]                          ! Temporary string for compilation
10030   INTEGER Xpos,Ypos                             ! Position counters
10040   String$="TX"&CHR$(34)&CHR$(1)                 ! Set up command with TP home code
10050   Ypos=Taby                                     ! Add required number of TP down codes
10060   WHILE Ypos>1
10070     String$=String$&CHR$(10)
10080     Ypos=Ypos-1
10090   END WHILE
10100   Xpos=Tabx                                     ! Add required number of TP right codes
10110   WHILE Xpos>1
10120     String$=String$&CHR$(7)
10130     Xpos=Xpos-1
10140   END WHILE
10150   String$=String$&Text$&CHR$(34)                ! Add text and close quotes
10160   OUTPUT @Analyser;String$                      ! Send complete command to 6500
10170   DEALLOCATE String$                            ! Discard compiled string
10180 SUBEND
```

Example 16 Keysrq

21. This routine sets up SRQ on front panel operation and waits for an interrupt to occur. The routine returns a numeric value indicating a key press, brightline movement or invalid SRQ condition.

Input requirements : NONE

| Output definitions : | Series 200 | HP85 |
|---|---|---|
| Front panel code | see below | K |

Sub-program calls : NONE

The Series 200 version of this routine is implemented as a function and should be called with :

Variable=FNKeysrq

The value returned will be in the range -1 to 40. A value of -1 indicates that the SRQ received was not generated by 6500. Values 0 to 38 represent front panel keys as defined in Appendix D. Values 39 and 40 indicate brightline left and right respectively.

*HP85 Sub-program*

```
1000 SUB "Keysrq" (K)
1010 COM A1
1020 K=-2 !                                          Preset code value
1030 OUTPUT A1 ;"SQ00010" !                          Set up 6500 SRQ mask
1040 ON INTR 7 GOSUB 1120 !                          Set up controller interrupts
1050 ENABLE INTR 7;8
1060 IF K<>-2 THEN 1080 !                            Wait for valid code
1070 GOTO 1060
1080 OUTPUT A1 ;"SQ00000" !                          Disable 6500 SRQ
1090 OFF INTR 7 !                                    and controller interrupts
1100 SUBEXIT !                                       Exit with code
1110 ! ------------------------------------------------- SRQ service routine
1120 STATUS 7,1 ; S !                                Reset interface interrupt
1130 S=SPOLL(A1) !                                   Fetch status byte from 6500
1140 IF NOT BIT(S,6) THEN K=-1 @ RETURN !            Check SRQ active
1150 S=S-64 !                                        Mask SRQ bit
1160 IF S<>1 THEN 1200
1170 OUTPUT A1 ;"RK" !                               If key-press
1180 ENTER A1 ; K !                                  Read key code
1190 RETURN !                                        and return
1200 IF S=2 THEN K=39 @ RETURN !                     If BL left set code=39
1210 IF S=3 THEN K=40 @ RETURN !                     If BL right set K=40
1220 K=-1 !                                          This should never occur
1230 RETURN
1240 SUBEND
```

*HP200 Sub-program*

```
10000 DEF FNKeysrq
10010   COM /A/ @Analyser
10020   Code=-2                                          ! Preset code value
10030   OUTPUT @Analyser;"SQ00010"                       ! Set up 6500 SRQ mask
10040   ON INTR 7 GOSUB Service                          ! Set up controller interrupts
10050   ENABLE INTR 7;2
10060   WHILE Code=-2                                    ! Wait for valid code
10070   END WHILE
10080   OUTPUT @Analyser;"SQ00000"                       ! Disable 6500 SRQ
10090   OFF INTR 7                                       ! and controller interrupts
10100   RETURN Code                                      ! Exit with code
10110 !-------------------------------------------------!
10120 Service:                                           ! SRQ service routine
10130 !-------------------------------------------------!
10140   Sbyte=SPOLL(@Analyser)                           ! Fetch status byte from 6500
10150   IF NOT BIT(Sbyte,6) THEN                         ! Check SRQ active
10160     Code=-1                                        ! Indicate no SRQ with negative code
10170     RETURN
10180   END IF
10190   Sbyte=Sbyte-64                                   ! Mask SRQ bit
10200   SELECT Sbyte                                     ! Examine status byte
10210   CASE 1                                           ! If key-press fetch key code
10220     OUTPUT @Analyser;"RK"
10230     ENTER @Analyser;Code
10240   CASE 2                                           ! If BL left set code=39
10250     Code=39
10260   CASE 3                                           ! If BL right set code=40
10270     Code=40
10280   CASE ELSE                                        ! This should never occur
10290     Code=-1
10300   END SELECT
10310   RETURN
10320 FNEND
```

Example 17 Plocal

22. This routine implements a 'pseudo-local' mode of operation where the instrument appears to be operating under local control, but is in face under remote control. This is achieved by intercepting all front panel operations with SRQ interrupts and simulating the operations performed with GPIB commands. The routine illustrates the use of the XK command. A key code value is supplied to the routine which is used as an exit code. For example if the value 16 was supplied control would revert to the main program when the LOCAL key was pressed. A status code is returned indicating exit due to a non-6500 interrupt.

23. A square block is written to the top left of the CRT to indicate 'pseudo-local' operation. When the exit code is recognized a housekeeping routine is performed to ensure that the instrument is left in a predictable state. In some cases the exit procedure involves waiting for valid data or the end of a plot function. During this waiting period the indicator is caused to flash. The indicator is removed on leaving the routine.

24. Due to speed limitations an HP85 version of this routine is not presented, although it is technically feasible to implement.

| Input requirements : | Series 200 |
|---|---|
| Exit code | Exit_code |
| **Output definitions :** | **Series 200** |
| Status code (0=normal, 1=non-6500 SRQ) | Exit_status |

Sub-program calls : NONE

The Exit code should be in the range 0-38 and represents a key code as defined in Appendix D.

25. This routine is not intended to be a solution to any specific system requirement but illustrates the extent to which the front panel SRQ facility may be used. Many useful additions could be made to the routine. Here are some examples :

Sending previously acquired normalization data when frequency limits are changed.

Switching off RF power when AUTO ZERO is pressed.

Performing automatic digital plots when PLOT is pressed.

Modifying CRT annotation.

Adding extra operator prompts and instructions.

Implementing 'menu' operation.

*HP200 Sub-program*

```
10000 SUB Plocal(Exit_code,Exit_status)
10010   COM /A/ @Analyser
10020   ALLOCATE Status$[24]                                    ! Temporary string for status
10030   INTEGER Sh_flag                                         ! Shift key flag
10040   Indic$="TX"&CHR$(34)&CHR$(1)&CHR$(127)&CHR$(34)         ! TX string for mode indicator
10050   Exit$=Indic$                                            ! TX string for exit indicator
10060   Exit$[5,5]=CHR$(255)
10070   Remov$=Indic$                                           ! TX string to remove indicator
10080   Remov$[5,5]=" "
10090   OUTPUT @Analyser;"SQ00010"                              ! Set up 6500 SRQ mask
10100   OUTPUT @Analyser;Indic$                                 ! Display indicator
10110   Sh_flag=0                                               ! Pre-set shift flag
10120   !-------------------------------------------------------!
10130   LOOP                                                    ! MAIN LOOP
10140     GOSUB Fetch_key                                       ! Fetch key code
10150   EXIT IF Key_code=Exit_code OR Key_code=-1               ! Leave if EXIT key pressed or non-6500 SRQ
10160     SELECT Key_code                                       ! Examine key code
10170     CASE -1
10180       Command$="0"                                        ! Dummy command if not 6500 SRQ
10190     CASE 20
10200       Sh_flag=NOT Sh_flag                                 ! Toggle shift flag
10210       Command$="SH"
10220     CASE ELSE
10230       Command$="XK"                                       ! Execute all other keys
10240     END SELECT
10250     IF Sh_flag AND Command$<>"SH" THEN
10260       Command$="SH"&Command$                              ' Ensure correct SHIFT operation
10270     END IF
10280     OUTPUT @Analyser USING "#,K";Command$                 ! Send command to 6500
10290     OUTPUT @Analyser;Indic$                               ! Re-write indicator
10300     IF Sh_flag THEN                                       ! Shift flag set contingency
10310       IF Command$<>"SH" THEN                              ! Commands reset SHIFT state
10320         Sh_flag=0
10330       ELSE
10340         OUTPUT @Analyser USING "#,K";"SH"                 ' Send SHIFT to retain mode
10350       END IF
10360     END IF
10370   END LOOP
10380   !-------------------------------------------------------! EXIT housekeeping.......
10390   OUTPUT @Analyser;Exit$                                  ! Change to flashing indicator
10400   OUTPUT @Analyser;"RX"
10410   ENTER @Analyser;Status$                                 ! Read status
10420   IF Status$[11,11]="7" OR Status$[14,14]="1" THEN        ! If in CALAID or PLOT menu
10430     OUTPUT @Analyser;"N"                                  ! Clear with NORMAL
10440   ELSE
10450     IF Status$[14,14]="2" THEN                            ! If PLOT routine active
10460       REPEAT                                              ! wait until finished
10470         WAIT .1
10480         OUTPUT @Analyser;"RX"
10490         ENTER @Analyser;Status$
10500       UNTIL Status$[14,14]="1"
10510       OUTPUT @Analyser;"N"                                ! Clear with normal
10520     ELSE
10530       IF Status$[5,5]="0" THEN                            ! Wait for valid data
10540         REPEAT
10550           WAIT .1
10560           OUTPUT @Analyser;"RX"
10570           ENTER @Analyser;Status$
10580         UNTIL Status$[5,5]="1"
10590       END IF
```

*HP200 Sub-program 2*

```
10600       IF Status$[8,8]="0" THEN                          ! If not in FREEZE
10610         IF Status$[10,10]="M" THEN                      ! Check for MEMORY display
10620           OUTPUT @Analyser;"N"                          ! Clear with NORMAL
10630         ELSE
10640           OUTPUT @Analyser;"FZFZ"                       ! Prompts cleared by toggling FREEZE
10650         END IF
10660       END IF
10670     END IF
10680   END IF
10690   OUTPUT @Analyser;"SQ00000"                            ! Reset 6500 SRQ mask
10700   OUTPUT @Analyser;Remov$                               ! Remove indicator
10710   DEALLOCATE Status$                                    ! Discard status
10720   IF Key_code=-1 THEN                                   ! Set EXIT status
10730     Exit_status=1
10740   ELSE
10750     Exit_status=0
10760   END IF
10770   SUBEXIT                                               ! Exit from routine
10780 !-------------------------------------------------------!
10790 Fetch_key:                                              ! Keyboard control routine
10800 !-------------------------------------------------------!
10810   Key_code=-2                                           ! Preset key code
10820   ON INTR 7 GOSUB Service                               ! Set up controller interrupts
10830   ENABLE INTR 7;2
10840   WHILE Key_code=-2                                     ! Wait for valid key code
10850   END WHILE
10860   OFF INTR 7                                            ! Disable interrupts
10870   RETURN                                                ! Return with key code
10880 !-------------------------------------------------------!
10890 Service:                                                ! SRQ service routine
10900 !-------------------------------------------------------!
10910   Sbyte=SPOLL(@Analyser)                                ! Fetch 6500 status byte
10920   Sbyte=Sbyte-64                                        ! Mask SRQ bit
10930   SELECT Sbyte                                          ! Examine status byte
10940   CASE 1                                                ! If key press
10950     OUTPUT @Analyser USING "#,K";"RK"                   ! Read key code
10960     ENTER @Analyser;Key_code
10970   CASE 2                                                ! If brightline left
10980     GOSUB Read_pos                                      ! Find current position
10990     IF Blpos>0 THEN OUTPUT @Analyser USING "#,K";"BL"   ! Execute BL if valid
11000     IF Sh_flag THEN OUTPUT @Analyser USING "#,K";"SH"   ! Re-assert SHIFT if necessary
11010   CASE 3                                                ! If brightline right
11020     GOSUB Read_pos                                      ! Find current position
11030     IF Blpos<421 THEN OUTPUT @Analyser USING "#,K";"BR" ! Execute BR if valid
11040     IF Sh_flag THEN OUTPUT @Analyser USING "#,K";"SH"   ! Re-assert SHIFT if necessary
11050   CASE ELSE                                             ! If non-6500 SRQ
11060     Key_code=-1                                         ! Return invalid KEY_CODE
11070   END SELECT
11080   IF Key_code<>-1 THEN ENABLE INTR 7                    ! Re-enable interrupts if valid key_code
11090   RETURN                                                ! and return
11100 !-------------------------------------------------------!
11110 Read_pos:                                               ! Read current brightline position
11120 !-------------------------------------------------------!
11130   OUTPUT @Analyser USING "#,K";"RP"
11140   ENTER @Analyser;Blpos
11150   RETURN
11160 SUBEND
```

## Binary/ASCII conversions

27. The binary and ASCII measurement transfers illustrated in examples 11-14 have different areas of application, but it may be desirable in some cases to convert from one format to the other. Two routines are presented which perform these conversions and will run on both the HP Series 200 and the HP85 computers.

The first example illustrates conversion from binary to ASCII. Previously acquired binary data is assumed to be in the string B$ and the target array is A which must be dimensioned with 422 elements (base 0).

The second example illustrates conversion from ASCII to binary. Previously acquired or generated ASCII data is assumed to be in the array A and the target string is B$ which must be dimensioned with 846 characters.

Essentially the routines are manipulations in 2's complement arithmetic. Note, however, that the sign bit is 0 for a negative value in 6500 internal format.

*Binary string to numeric array conversion*

```
1000  FOR I=0 TO 421
1010    M=NUM(B$[I*2+4,I*2+4])
1020    L=NUM(B$[I*2+3,I*2+3])
1030    N=L+M*256
1040    IF N>32767 THEN N=N-65536
1050    A(I)=-N/256
1060  NEXT I
```

*Numeric array to binary string conversion*

```
1000  B$="$I"
1010  FOR I=0 TO 421
1020    N=-INT(A(I)*256+.5)
1030    IF N<0 THEN N=N+65536
1040    M=N DIV 256
1050    L=N MOD 256
1060    B$=B$&CHR$(L)&CHR$(M)
1070  NEXT I
```

Appendix A

# 6500 FIRMWARE* ISSUE 5 AND PRE ISSUE 5 COMPARISON

1. The dB REL commands (SH8 and SH9) now require the E terminator following an optional numeric argument. With pre-issue 5 versions these commands did not use any arguments.

2. The brightline frequency value output following the RF command is now in GHz rather than Hz. This gives compatibility with the frequency input commands, which all assume GHz units.

3. The brightline amplitude values output following the RA, RB, or RR commands are now rounded to two decimal places. With pre-issue 5 versions three decimal places were output, despite the resolution limitation of 1/256 dB.

4. The interpretation of bits 0-3 of the serial poll status byte has changed. With pre-issue 5 versions decimal values of 1-9 were interpreted as numeric key presses 1-9. The new interpretation (see SRQ FACILITIES - section 32) coupled with the RK command allows SRQ interrupts on any key press or brightline movement.

5. The following commands are not available with pre-issue 5 versions :

| | | | | | |
|---|---|---|---|---|---|
| RMA | RMB | RMR | RYA | RYB | RYR |
| SHRMA | SHRMB | SHRMR | SHRYA | SHRYB | SHRYR |
| WMA | WMB | WMR | WYA | WYB | WYR |
| SHWMA | SHWMB | SHWMR | SHWYA | SHWYB | SHWYR |
| RDA | RDB | RDR | RDF | RI | WI |
| RX | RMK | RP | RK | XK | DA |
| TX | DPA | DPB | DPG | DPL | |

*Firmware is the term given to the instrument's internally fixed operating system.

Appendix B

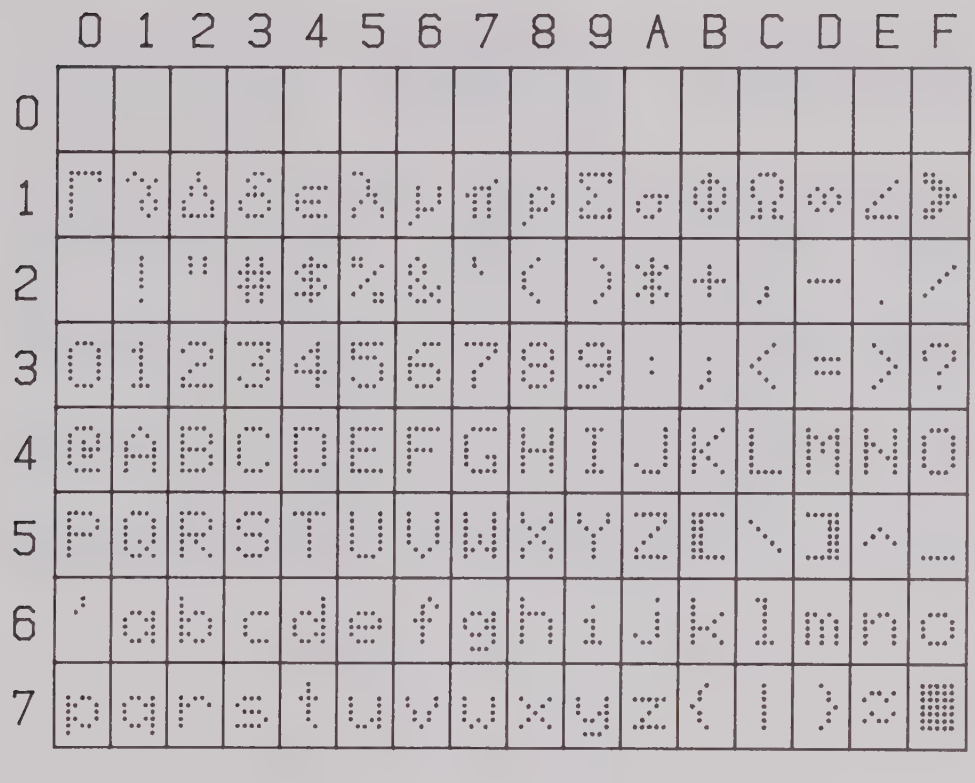
# GPIB ERROR CODES

| *Code* | *Meaning* |
|---|---|
| 31 | Attempt to move brightline outside valid area (positions 0-421) using the BL, BR or BP commands. |
| 50* | ASCII data in WM transfer outside valid range (+/-99.99). |
| 51* | Format error - illegal character in WM ASCII transfer data string. |
| 52* | Premature termination of WY, WM or WI transfer. |
| 53* | No termination of WY, WM or WI transfer. |
| 54* | Listen buffer overflow (may be caused by absence of separators between ASCII values in WM transfer). |
| 55 | Argument to RI or WI command outside valid range (1-9). |
| 56 | Attempt to read from a 'null' store with RI. |
| 57* | Format error on WI transfer block header. |
| 58* | Checksum error on WI transfer. |
| 60 | Attempt to read an HPGL plotter control string (DPA, DPB, DPG or DPL commands) when in CALAID mode. |

*These commands cause the operation to be aborted until receipt of a valid SR2/3 terminator [CR]LF or EOI. See IEEE 728-1982 for full definition of terminators.

Appendix C

# 6500 CHARACTER SET AND TEXT CONTROL CODES



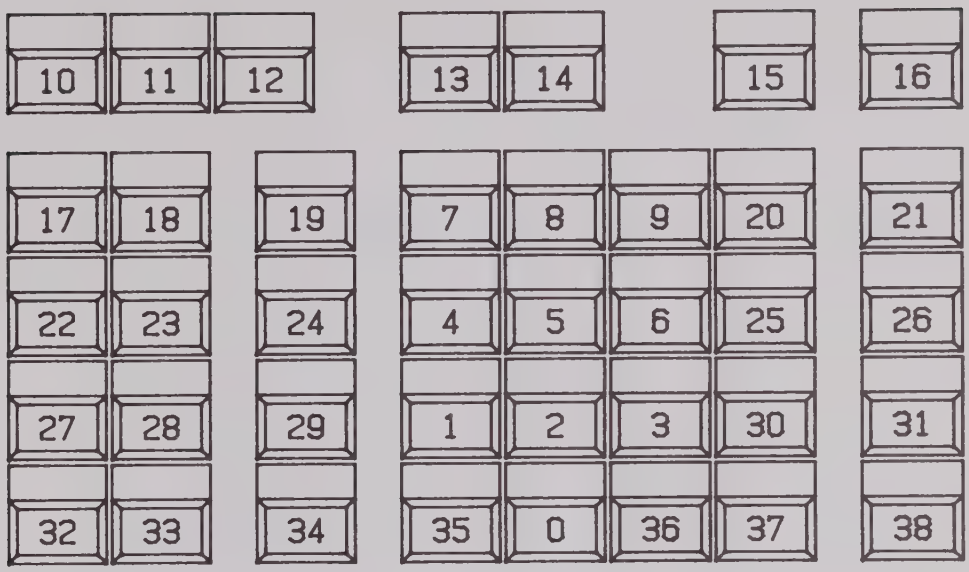

## CONTROL CODES

Character codes 0 to 15 are not displayed but are used as control codes in the following manner :

| *Code* | *Function (TP = Text Pointer)* |
|---|---|
| 0 | Ignored. |
| 1 | Move TP to top left of screen (Row 1, Column 1). |
| 2 | Clear current row and move TP to column 1. |
| 3 | Clear text from TP to end of current row. |
| 4 | Set TAB at current TP column. |
| 5 | Reset TAB at current TP column. |
| 6 | Clear all tabs. |
| 7 | Move TP one position right unless at column 40. |
| 8 | Move TP one position left unless at column 1. |
| 9 | Move TP to next TAB position or column 40. |
| 10 | Move TP one position down unless on row 24. |
| 11 | Move TP one position up unless on row 1. |
| 12 | Clear all text and move TP to top left of screen. |
| 13 | Move TP to column 1 of current row. |
| 14 | Enable flash mode for subsequent characters. |
| 15 | Disable flash mode. |

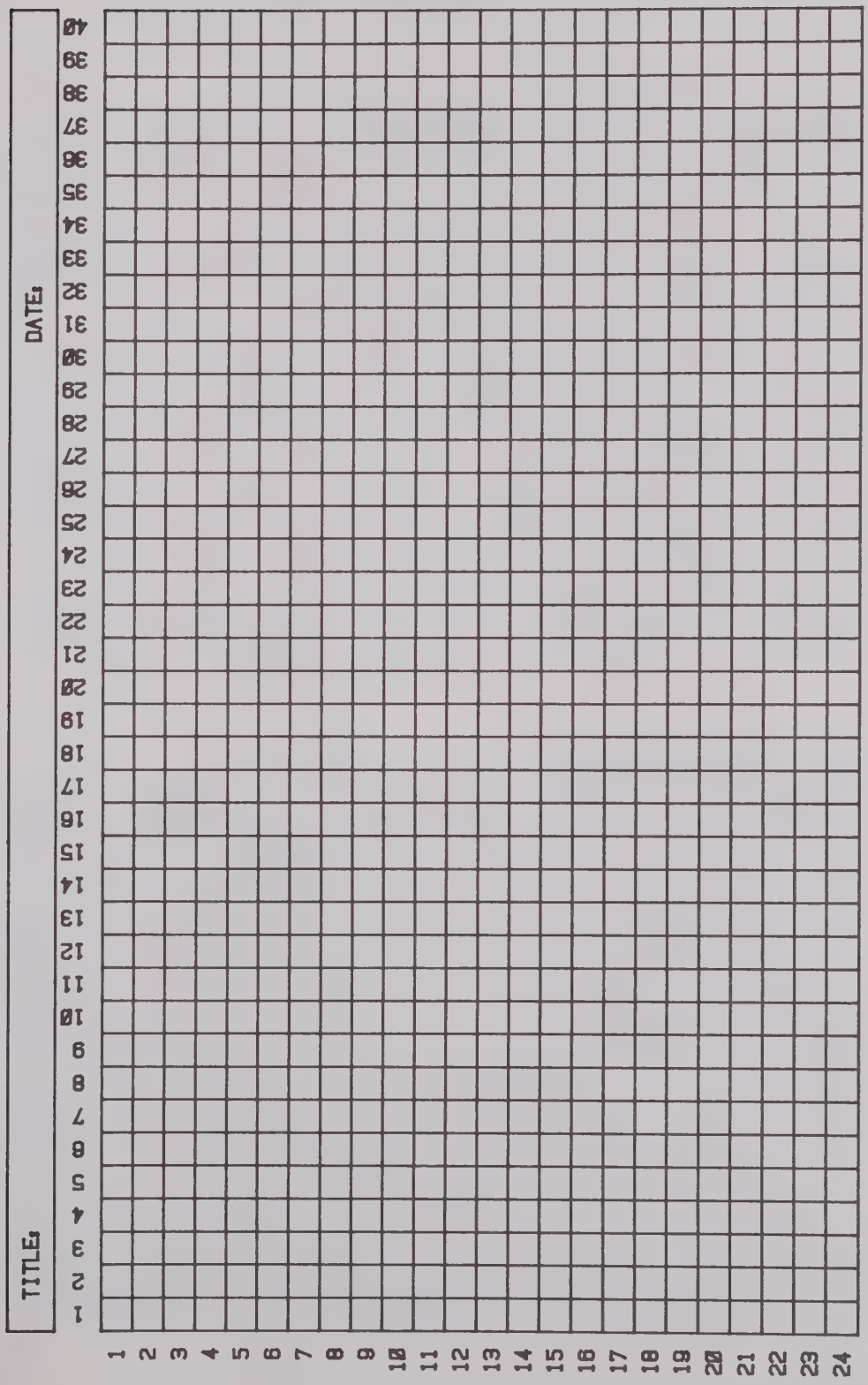
Appendix D

# KEYBOARD CODES RETURNED BY RK COMMAND



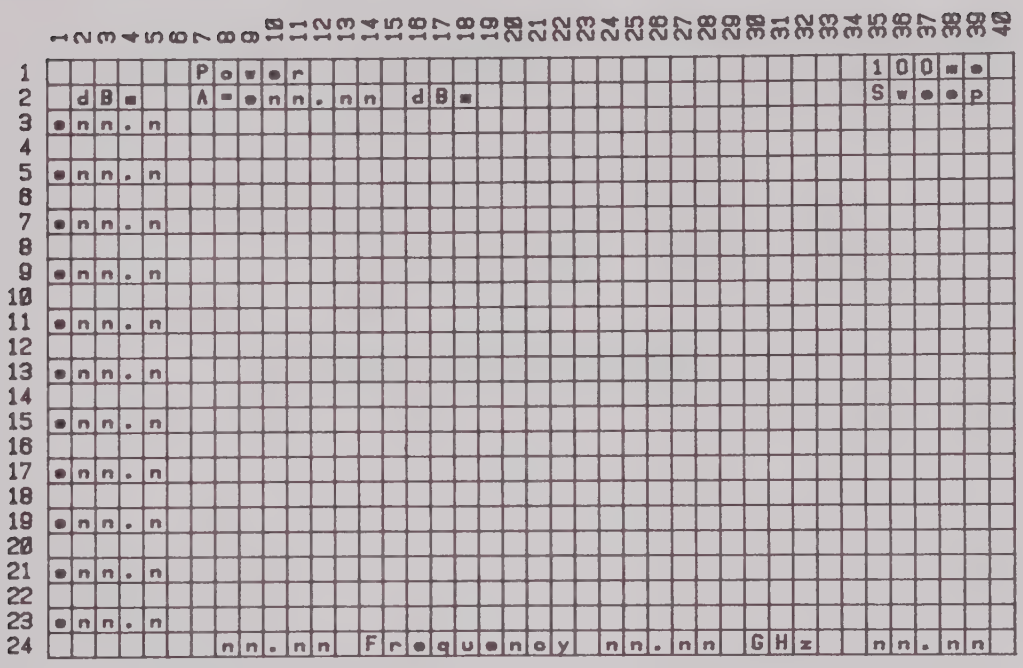
Appendix E

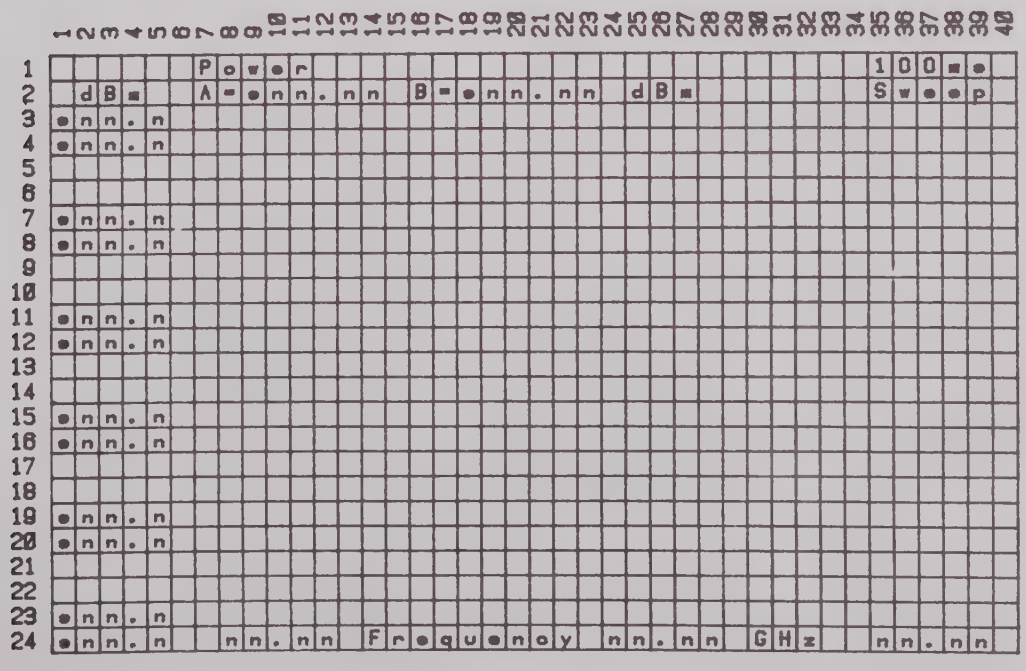
# 6500 CRT LAYOUT CHART



Appendix F

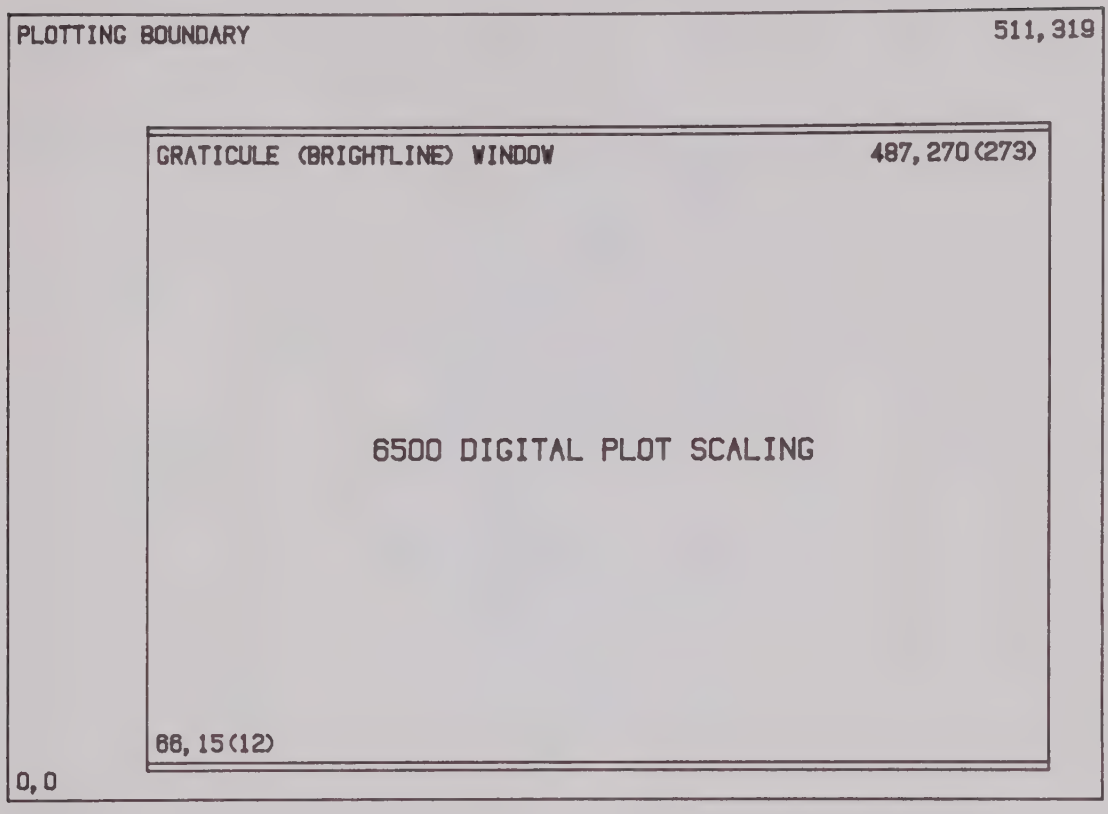
# STANDARD ALPHANUMERIC LAYOUTS FOR SINGLE AND DUAL CHANNEL DISPLAYS





Appendix G

# DIGITAL PLOTTER OUTPUT INFORMATION




The plotting area is scaled as 512 x 320 user units.

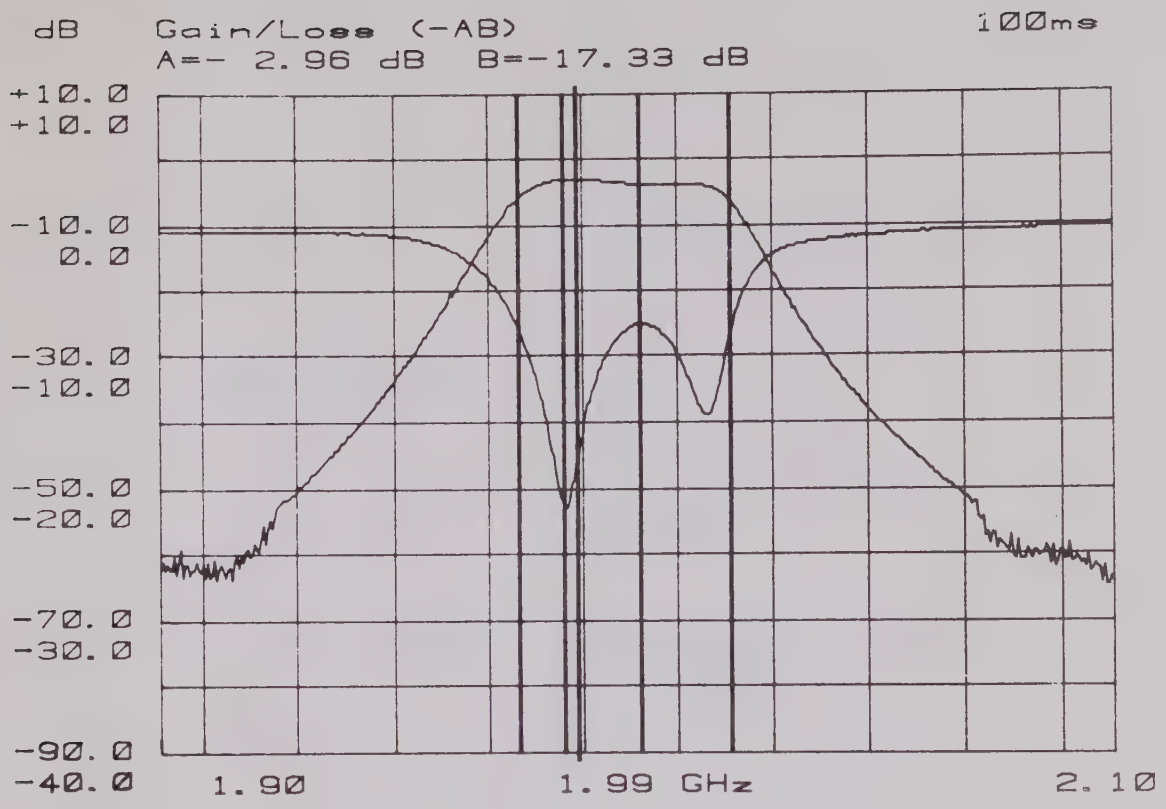
The character size is set to: 1.456% of vertical scale
2.000% of horizontal scale.

Starting co-ordinates of labels:

| | | |
|---|---|---|
| Units | 0,293 | |
| Sweep Speed | 429,293 | |
| Start Frequency | 66, 0 | |
| Brightline Frequency | 221, 0 | |
| Stop Frequency | 440, 0 | |
| Datum 1 | 0,268 | |
| Datum 2 | 0,256 | (Dual channel only) |
| Top Line | 66,293 | |
| Second Line | 66,281 | |

Appendix H

# SAMPLE DIGITAL PLOT



**MARCONI INSTRUMENTS LIMITED**

| | |
|---|---|
| **SERVICE DIVISION** | The Airport<br>LUTON<br>Bedfordshire LU2 9NS<br>Telephone: (0582) 33866<br>Telex: 825248 |
| **HEAD OFFICE** | Longacres<br>ST ALBANS<br>Hertfordshire AL4 0JN<br>Telephone: (0727) 59292<br>Telex: 23350<br>Fax: (0727) 57481(Gp.3) |
| **MICROWAVE PRODUCTS (Service)** | PO Box 10<br>Gunnels Wood Road<br>STEVENAGE<br>Hertfordshire SG1 2AU<br>Telephone: (0438) 312311<br>Telex: 82159 |
| **FRANCE** | MARCONI INSTRUMENTS<br>Parc d'activités du Canal<br>18 Rue du Plessis-Briard<br>Courcouronnes<br>91023 EVRY<br>Telephone: (6) 077 90 66<br>Telex: 690428<br>Fax: (6) 077 69 22 |
| **WEST GERMANY** | MARCONI MESSTECHNIK GmbH<br>Postfach 153<br>Landsberger Strasse 65<br>8034 GERMERING<br>Telephone: (089) 84 50 85<br>Telex: 5212642<br>Fax: (089) 8419142(Gp.3) |
| **USA** | MARCONI INSTRUMENTS<br>3 Pearl Court<br>Allendale Park<br>ALLENDALE<br>New Jersey 07401<br>Telephone: (201) 934 9050<br>Telex: 6853077 MIUSA<br>Twx: (710) 991 9752<br>Fax: (201) 934 9229 |

MA25529